2

MENTATION PAGE

Form Approved
OMB No. 0704-0188

estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, g and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this g this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

REPORT DATE

3. REPORT TYPE AND DATES COVERED
FINAL 15 Sep 91 - 14 Jan 92

4. TITLE AND SUBTITLE
"WAVELET SIGNAL PROCESSING FOR TRANSIENT FEATURE EXTRACTION" (U)

5. FUNDING NUMBERS
61102F
2500/00

6. AUTHOR(S)
Dr Karl B. Jagler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Aware, Inc.
One Memorial Drive
Cambridge, MA 02142-1301

8. PERFORMING ORGANIZATION REPORT NUMBER
AEOSR-TR- 92 0402
AD920315

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
Bldg 410
Bolling AFB DC 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-91-C-0089

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release;
Distribution unlimited

12b. DISTRIBUTION CODE
UL

13. ABSTRACT *(Maximum 200 words)*

Research was conducted to evaluate the feasibility of applying Wavelets and Wavelet Transform methods to transient signal feature extraction problems. Wavelet transform techniques were developed to extract low dimensional feature data that allowed a simple classification scheme to easily separate the various signals of interest. Additional development of these techniques will lead to robust feature extraction methods for transient signals.

14. SUBJECT TERMS

15. NUMBER OF PAGES
81

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

AWARE, INC.
One Memorial Drive
Cambridge, MA 02142-1301 USA
TEL 617-577-1700 FAX 617-577-1710

®

Wavelet Signal Processing for Transient Feature Extraction

Final Report

prepared for
Air Force Office of Scientifc Research
under contract F49620-C-0089

AD920315

Wavelet Signal Processing
for
Transient Feature Extraction

Final Report

Aware, Inc.
One Memorial Drive
Cambridge, MA 02142
617-577-1700 617-577-1710 FAX

15 March 1992
AD920315

prepared for
Air Force Office of Scientific Research
under contract F49620-91-C-0089

Executive Summary

Research was conducted to evaluate the feasibility of applying Wavelets and Wavelet Transform methods to transient signal feature extraction problems. Wavelet transform techniques were developed to extract low dimensional feature data that allowed a simple classification scheme to easily separate the various signals of interest. Additional development of these techniques will lead to robust feature extraction methods for transient signals.

Detailed study results are presented in section 3.3 on page 53.

Contents

List of Figures

Chapter 1

Introduction and Overview

This final report details the results of research conducted into the feasibility of applying Wavelets and Wavelet Transform methods to transient signal feature extraction problems. This report contains both theoretical and experimental results of studies conducted with mechanical transient data. This work was performed under contract number F49620-91-C-0089 for the Air Force Office of Scientific Research (AFOSR.)

Aware, Inc. was tasked to perform the following activities:

1. **Selection of Wavelet Basis Functions and Transform Topologies.** Define and develop wavelet transforms which will provide time-frequency feature localization methods for the types of mechanical transient signals provided for analysis;

2. **Analysis and Characterization of Signal Features.** Measure the transient feature extraction characteristics of wavelet-based signal processing algorithms;

3. **Prototype Algorithm Development.** Select candidate detection and classification features and develop a prototype algorithm for that automatic detection and classification of the transient signals of interest.

Compactly Supported Wavelets

The reason for conducting this study is the remarkable results being obtained through the application of ***compactly supported***[1] ***wavelets*** to transient signal processing. Compactly supported wavelets are a class of mathematical functions that were discovered in 1986. Some of the particular advantages of wavelet signal processing methods are:

- Wavelet transforms are computationally efficient. The number of arithmetic operations required to perform a wavelet transform is **linearly** proportional to the number of input data points. The computational complexity of the more traditional *Fast Fourier Transform* (FFT) is proportional to the number of input data points times the logarithm (base 2) of the number of input data points ($O(N \log_2 N)$). For large problems, the wavelet methods require only a fraction of the number of operations required by the traditional methods. This advantage increases as the problem size increases. In addition, wavelet transform algorithms can be directly implemented in Very Large Scale Integration (VLSI) logic devices, and they are fully parallelizable.

- Wavelet transform methods can analyze signals in both the time and frequency domains. The relative resolution of the time and frequency components can be flexibly adapted to the problem at hand. The selection of the appropriate time-frequency resolution can be done upfront at system design time or it can be accomplished with real-time adaptive algorithms. The traditional Fourier transform suffers from very poor (or nonexistent) time resolution. This particularly limits its usefulness in the analysis of time-limited (i.e., transient) signals. There have been attempts to modify the Fourier technique in various ways to overcome this limitation, but all of the methods introduce some additional complexities and compromises. Wavelet methods offer a very natural means to perform time-frequency signal analysis.

- Wavelets provide the flexibility to choose a particular wavelet function that is "customized" to the specific application. This is possible since

[1] Compactly supported means that the functions are identically zero outside a finite interval.

compactly supported wavelets are an *infinite family* of complete orthogonal basis functions. This flexibility to choose basis functions can not be matched with the Fourier transform for it uses only a *single* set of basis functions – the complex exponentials (i.e., the sine and cosine functions.)

Report Overview

This report begins with an introduction to *wavelet phase space.* This is the "playing field" on which we develop the concepts of wavelet signal processing. The development is intuitive with the rigorous mathematical details provided in the appendices. The basic concepts of wavelet signal analysis are developed and contrasted to traditional Fourier analysis methods. This is followed by a discussion of the computational characteristics and complexity of wavelet methods. Next the general transient signal processing problem is introduced and the applicability of wavelet signal processing methods to various parts of the problem is discussed. This is followed by a discussion of the results obtained with the specific set of signals which were studied. Excellent signal separation was demonstrated with the use of low dimensional wavelet extracted features in a simple classifier design.

To assist the reader in locating the particular sections of this report which address the contractually required items, the following guide is provided:

1. The selection of wavelet basis functions and transform topologies is discussed in section 3.2. Wavelet transforms are defined and developed on an intuitive level in chapter 2. A rigorous mathematical development of wavelets and wavelet transforms is presented in appendix B;

2. The analysis and characterization of signal features extracted via wavelet methods as well as the design of prototype algorithms are presented in section 3.3.

Chapter 2

Wavelet Signal Processing

This chapter introduces the concepts of signal, signal processing, mathematical transforms, phase space representation of signals, the wavelet transform and wavelet signal processing methods. The mathematical details of wavelets and wavelet transforms are rigorously discussed in appendix B. The development here is intuitive with the emphasis on explaining "what" a wavelet transform does rather than "how" it works or "why" it is mathematically correct. The Fourier transform will be introduced because it is the signal processing technique most frequently used today. The objective is to provide an overview of the capabilities of wavelet transform methods and an understanding of their relative strengths and weaknesses compared with Fourier methods.

2.1 Signals and Signal Processing

A *signal* is something that conveys information. In this discussion, signals will usually consist of a series of measurements of some physical quantity such as voltage or force. The information conveyed is often about the occurrence of events or the state and behavior of a physical system.

The discussion will be limited to the class of signals that could be represented in a digital computer. These must be *discrete* signals, i.e., signals that are represented by a sequence of numbers whose values are of finite precision i.e., representable by a finite number of bits. The signals may be samples of a continuous quantity (i.e., measurements of the value of a physical quantity with a sensor) or values from a completely discrete process (e.g., counting the occurrences of some event.) Due to their discrete nature, these signals contain *finite energy*. The energy of a discrete signal is calculated by summing the squares of its values. Let $s_k, k = (0, 1, 2 \ldots)$ represent the sequence of values of a discrete signal. The energy of the signal E is

$$E = \sum_k s_k^2 \,. \tag{2.1}$$

Signal Processing is a collection of methods used to change the representation of a signal into a form that allows the information to be interpreted more easily. Signal processing is used to locate events or identify system *modes* or other characteristic behaviors. Signal processing methods are used to separate signals from noise or separate signals from each other.

Noise is extraneous energy that is combined with a signal. It interferes with the ability to interpret the information being conveyed by the signal. Noise has many sources, both natural and man-made. It can have characteristics that are either random or deterministic or a combination of both. Whether energy represents noise or signal sometimes depends on the interests of the investigator. For example, in speech signal processing, speaker-dependent characteristics of the acoustic speech waveform are "noise" if it is desired to extract the meaning of the utterance, but the semantic characteristics are "noise" if the problem is to identify the speaker.

A simple example of a signal processing operation is calculating the average or mean of a sequence of values. A signal, consisting of a sequence of 'noisy' temperature measurements, can be "signal processed" by averaging a large number of samples to separate the steady temperature estimate from the randomly fluctuating noise.

Signal processing methods vary in the amount of computation required to perform them. We measure the computational complexity of various methods by establishing a relationship between the number of input data points to be processed and the number of arithmetic calculations required to perform the method. In the previous example of the averaging operation, the complexity is of *linear order* because the number of operations required to perform the averaging method is simply proportional to the number of data points to be averaged. This is written as $O(n)$ (the operation is said to have a computational complexity of *order n*.) A method that required a number of arithmetic operations proportional to the square of the number of input points would be of *quadratic order* and written as $O(n^2)$. This notation is useful for comparing the relative amount of processing required by different methods for large amounts of input data. It represents the "trend" or asymptotic limit for large problems. The comparison of computational complexity for a particular small processing problem usually requires a more detailed analysis than simply examining the "trend" for large problem sizes.

2.2 An Introduction to Transforms, Basis Functions and Phase Space Representations

A *transformation* (or transform) is a mathematical process which changes the representation of a signal. Transforms are used in signal processing to extract information and separate signals from noise. They are described by properties that characterize their actions. The mathematical theory of transformations includes both *continuous* and *discrete* transforms. We will only employ discrete transforms since our signals are discrete. Several important transform properties will be defined to provide a vocabulary for the discussion of transformation techniques.

Transformations are called *invertible* if they are reversible. The process (or transformation) that reverses the action of a particular transformation is called the *inverse* of the transformation.

Energy-preserving transformations conserve signal energy. The formula for computing the energy contained in a signal was defined in the previous section by equation (2.1). An energy-preserving transformation may change the representation of a signal dramatically, but the energy of the input signal and the energy of the transformed output will always be equal. Non-energy-preserving transformations either lose energy or produce extra energy in the transformation process.

A special type of transformation is called a *linear* transformation. For a linear transformation, the results of transforming two signals and then adding the (transformed) signals together **is exactly the same as** adding the two signals together first and then transforming the sum (of the two signals). The results of signal processing methods that use linear transformations are easier to analyze than nonlinear methods because a common type of noise (typically found in electronic environments) is additive and in this case the action on the signal can be separated from the action on the noise since the two actions do not depend on each other.

Transformations may be *block* transforms or *stream processing* transforms. Block transform methods process the input data in "blocks" (or groups of a fixed size.) The output is also a "block", not necessarily the same size as the input block. The choice of block size affects the ability to *resolve* signal features larger than a block or smaller than a fraction of the block size. Block methods have difficulties resolving a signal feature that is split across two adjacent input blocks. We note that block transform algorithms have been designed to process a continuous stream of input data. These methods implicitly divide the input data into segments and the effects of blocking are always present in the output.

Stream processing transforms generate one or more output streams from an input stream. Such a method can be visualized as a "pipe", taking in data at one end and producing output at the other end. These methods can process blocks of data, which are just "streams" with a beginning and end.

To illustrate these properties we return to the averaging operation discussed in the previous section. The "averaging" transformation is a discrete transformation because it takes a discrete sequence of numbers and converts it into a discrete value (which might have to be rounded to a given precision.) Averaging is not invertible as many different sequences of input values could produce the same average value and there is no way of reversing the operation to obtain the correct input sequence. The transformation is not energy-preserving because the sum of the squares of the inputs does not equal the square of the average value. The transform is a linear transformation because the sum of the averages of two sequences does equal the average of the sum (element by element!) of the input sequences. This transformation can be implemented as either a block or stream process. Data could be processed in fixed blocks, producing one average value for each input block or it could be used to generate a "moving average", producing a series of outputs that represent the average value of a "window" that is moved across the input data.

The behavior of a system can often be understood in terms of *modes* or characteristic ways that a system operates or performs. A system may have a few or a very large number of distinct behaviors that can occur individually or in combinations. Consider the sound that a piano creates. It can produce single notes, combinations of notes or no notes at all. The notes can be played loudly or softly as well as rapidly or slowly. They could be separated in time or overlapping each other.

In this example the characteristic "modes" are the set of complicated vibration patterns produced by piano strings when they are struck. There is one "characteristic mode" per key, for a total of 88 "modes."' One "analysis" of piano sound seeks to determine that keys were pressed at what time and how hard they were pressed. We are going to define a "piano transform" to perform the "signal processing" part of this analysis procedure. The input to our "piano transform" is (a stream of) piano sounds and the output is 88 streams of numbers that represent how loud a "mode" (or key) is as a function of time. Zero values mean that a key is not pressed (said another way, the *amplitude* of the "mode" is zero.) Non-zero values represent how loud a "mode" is during some interval. The loudness of a note is the *amplitude* of the mode. This information could be displayed as 88 graphs, each stacked up above the other, tracing out how hard each key was pressed over time.

We are going to perform the "piano transform" on a computer with the following algorithm:

1. Create a mathematical description of the vibrating piano string for each "mode" (key) and store it for future reference;

2. Read in a section[1] of the signal and compare each stored reference (from the step above) to this section of the signal. Compute a match (or correlation) score that is proportional to how well the signal matches the reference.

3. Output the match score for each "mode";

4. Read in another section of signal and repeat the previous two steps.

This principle of comparing a known reference to a signal is conceptually how a transform method works. The mathematical representations of the modes (the references) are called ***basis functions.*** The matching scores or correlations are called the ***transform coefficients*** (or ***coordinates***) of a signal with respect to the basis functions. Basis functions are selected to model the modes or behavior(s) of the system being analyzed.

[1] We want to stress the concept that more than a single point must be considered.

A collection of basis functions is called a **basis**. If the basis has enough functions to represent all of the possible signals of a system, it is called a *complete* basis. Furthermore, if each the basis functions represents information that is *independent and uncorrelated*, the basis functions are called *orthogonal*[2]. Orthogonality is important and desirable because orthogonal basis functions have no redundancy in their transform representation.

Returning to the piano example, the basis functions (representations of the vibrating strings) form a complete and orthogonal basis for piano sounds. These basis functions have a *scale*[3] relationship between them. There is a constant *ratio*[4] between the frequencies of adjacent modes (keys.) Notes that are separated by an *octave* have a ratio of $2:1$ of their frequencies. The strings in a piano vary in length depending on the frequency of the note they produce. Low notes require long strings and high notes are produced by short strings. In fact, the lengths of the strings have the same ratios between them as the frequencies.

We have discussed the length of the strings because we are going to make the length of the "piano" basis functions vary the same way as the string lengths. This will affect the way we compute the "piano transform."

Imagine that we plotted all the "piano" basis functions at the same scale. The functions for the low notes are much longer than the functions for the high notes. This is because low note "modes" vibrate slower and have their sound waves spread across more signal values than high note "modes." To make this a little more concrete, let us say that a low note basis function has length 100 and a high note basis function has length 25 (These notes differ by two octaves since $100 = 2^2 \times 25$.) If 200 samples of piano sound are input to the piano transform for processing, only two sub-sections of the signal can be compared against the length 100 low note functions while 8 sub-sections of signal can be compared to the high note function of length 25. The long functions (which represent phenomena that vary slowly) have a coarse time resolution that is matched to the slow rate of variation. The short functions have correspondingly finer time resolution matched to the faster rate of vibration. This relationship between time resolution and how rapidly information changes is central to understanding the wavelet transform. Signal

[2] This definition of orthogonality is perhaps oversimplified, but will suffice for this example.

[3] Do not confuse this with a musical scale, though the ideas are closely related.

[4] In this example, the ratio is $\sqrt[12]{2}:1$ i.e., 12 notes span an octave.

features that vary over small scales (short distances or short time intervals) can be located precisely in time while features that vary over large scales can only be located with a correspondingly coarser time resolution.

Compactly supported wavelets are a complete and orthogonal set of basis functions for the set of *all* finite energy discrete signals. The wavelet transform is invertible, energy-preserving and linear. The wavelet transform is a stream processing method that analyzes both continuous streams of input data and blocks of data. A (multiplier 2) wavelet **basis** consists of a *scaling function*, a *basic wavelet* and a collection of smaller wavelets at reduced *scales.* The smaller wavelets are created by "shrinking" the basic wavelet by factors of 2 and shifting (or translating) them by scaled integer distances. Thus the collections of smaller wavelets are 1/2 and 1/4 and 1/8 (and so on) the size of the basic wavelet. Wavelet basis functions are all related by multiples of the constant ratio (2 : 1). The basic wavelet is computed from the scaling function. The selection of the scaling function determines all the remaining basis functions. A remarkable fact is that there are an infinite number of scaling functions, each of which defines a complete wavelet basis. This provides tremendous flexibility in selecting basis functions that are appropriate for different systems.

The time resolution of each wavelet is proportional to its length or duration, smaller wavelets having finer time resolution. Conversely, as the length of the wavelet increases, its resolution in scale (or frequency) gets smaller. The trade-off between resolution in time and resolution in scale is discussed in detail in the next two sections and will not be developed further in this section.

In general, the computational complexity of wavelet methods is $O(n)$. The efficiency is the direct result of the simplicity of the wavelet transform process. The process starts by separating the signal information at the smallest scale from information at all the larger scales. The output of the first stage is processed again by the same method and is repeated for each successive scale. This *recursive* structure reduces the amount of data to be processed at each successive level by a factor of two that reduces the computational cost for each successive transform level. Information which varies rapidly over just a few data points is separated from information that varies over many points. The procedure is stopped at the largest scale of interest.

The outputs of thc wavelet transform are coefficients that represent the similarity of the signal (as a function of time) to the wavelets at different scales. The output of a wavelet transform can be plotted on a grid that has time on the x-axis (or t-axis) and scale on the y-axis. This grid is called *phase space* and is used to graphically display the relationships between signal information at different scales.

We have used the piano example to introduce the basic ideas required to intuitively understand how transform methods work and what a wavelet transform does. We will now discuss the Fourier transform and illustrate how it is different from the wavelet transform.

The Fourier transform is invertible, energy-preserving and linear. The Fourier basis functions are orthogonal and complete for the entire set of finite energy discrete signals.

The Fourier transform separates signal information by frequency. The Fourier transform is a block transform which requires an *a priori* choice of block size. The discrete Fourier basis functions are uniformly sampled constant frequency sine and cosine functions each of which persists as long as the block size. The basis functions are uniformly spaced in frequency from zero to one-half the block size. The frequencies are separated by a constant interval rather than a constant ratio. Since all the basis functions are as long as the input data block, they all have the same (lack of) time resolution. The output of a Fourier transform contains information about how the energy in the signal is distributed among the frequencies in the signal. However information about how the energy is distributed in time, about *when* it occurred, is not available in the Fourier transform representation. All that can be inferred is that the frequency was present somewhere in the block and for what fraction of the signal energy it accounted. The computational complexity of the Fast Fourier Transform (for the commonly used Cooley-Tukey algorithm) is $O(n \log_2 n)$.

In summary, the primary difference between the Fourier transform and the wavelet transform is in how each separates signal information between time and scale[5].

The Fourier transform is a block transform that separates signal information into uniformly spaced frequency components. The Fourier transform has fine frequency resolution and a complete lack of time resolution. Larger

[5] Scale is related to wavelength which is defined as 1/frequency for periodic functions. For periodic signals, there is a close correspondence between scale and wavelength and therefore, frequency. The behavior of aperiodic signals can be analyzed in terms of large and small scale variations that are not periodic and do not have a well defined wavelength.

block sizes increase the range of frequencies that the Fourier method can resolve, but further decreases the time information available from a signal.

Wavelet transforms separate signal information by scale and time. There are an infinite number of wavelet basis from which an appropriate basis can be selected. The transform information is defined by a constant scaling relationship with the time resolution proportional to the scale. The computational complexity of the wavelet transform is $O(n)$ less than the Fourier transform $O(n \log_2 n)$.

The next two sections will refine a few of the mathematical concepts related to wavelet basis functions and the mechanics of the wavelet transform process. At the expense of a few details and mathematical precision, the reader may skip to section 2.3.3 or even 2.4 without loss of continuity.

2.3 The Scaling Function, Wavelets, and the Wavelet Transform

Wavelet methods separate the components of a signal by *scale*. Here *scale* means a *level of detail or resolution*. Its significance can be expressed in terms of a "correlation length", or the length over which data at a given *scale* tends to vary significantly. *Scale* and *frequency* are logically independent concepts. In many circumstances there is a relationship between scale and frequency that arises from the specific details of the signal processing problem. In these cases, instead of referring to low frequency and high frequency, one refers to large scale and small scale. The representation of a signal in terms of scale - the wavelet representation - separates the signal into components that are independent and uncorrelated (i.e., orthogonal), yet each has a well defined scale-specific level of detail.

With a wavelet transform, both the time scale resolution, or "correlation length", and the frequency resolution vary logarithmically. Wavelet techniques divide the spectrum of a signal into equal width bands on a log-frequency scale; they provide an octave band decomposition of the signal. This scale-oriented approach provides for finer frequency resolution in low frequency, large scale bands, and lesser frequency resolution at high frequencies and small scales. Correspondingly, the wavelet decomposition provides for coarse temporal resolution at low frequencies (since changes happen slowly at low frequencies), and fine temporal resolution at high frequencies, where rapid changes may occur.

By focusing on scale resolution rather than on frequency or time alone, the wavelet technique considers the reciprocal relationship between time and frequency (or any type of structure that is expressed across multiple data points). To identify or locate the position of a particular shape, such as an oscillation, in a set of data, one must look for relationships among the data values; a structure or an oscillation exists only across a set of data, and not in a single point value. A number has no frequency, no structure. Conversely, properties that exist only across a set of data cannot be said to have a particular location within that set. Wavelet signal representation techniques take this trade-off into account by allowing small sets of data to be combined and correlated to derive structural or shape information about that subset, without requiring a complete transformation of the signal into a particular type of structural information, the way a Fourier transform does. One is allowed to exchange a small amount of temporal resolution for a small amount of scale information, a pay-as-you-go system.

It is no accident that these properties are reflected in the characteristics of many natural signals. Signals with time varying characteristics, like speech, music, seismic signals and underwater acoustic signals are all best analyzed by a system capable of resolving both frequency and time. Furthermore, many signal-producing phenomena have octave band structure due to the presence of harmonics within the signal. Transient events also respond well to multiple scale analysis in that the identification of precisely located phenomena, such as the sharp onset of a signal, requires the ability to resolve its location in time at a very fine scale, while the characteristics of later, more persistent, parts of the signal may require the ability to identify larger scale structures.

2.3.1 The Scaling Function

The scaling function is at the core of any wavelet based representation of a signal. We will discuss only *compactly supported wavelets* in this report. The scaling function has three essential properties. The first is that it is *compactly supported.* This means that the scaling function is exactly zero outside a bounded region of the real line. The scaling function is only locally non-zero. The second essential property is that the scaling function is orthogonal to integer translates of itself. The importance of this will become clear a little later. The third property is that the scaling function is intimately related to smaller *scaled* versions of itself. This relationship is expressed concisely by the *scaling equation*:

$$\varphi(x) = \sum_{k=0}^{N-1} a_k \varphi(2x - k) \tag{2.2}$$

where $\varphi(x)$ is the scaling function. The function $\varphi(2x)$ is a smaller, scaled down (by a factor of two), version of $\varphi(x)$. The scaling equation states that $\varphi(x)$ is equal to a weighted sum of these smaller shifted versions of itself. The numbers a_k, of which only finitely many (N, which is an even number here) are non-zero, are called the *scaling coefficients.* N is the *size* of the wavelet system. The *support* of φ, the region on which it is non-zero, is the interval $[0, N-1]$. The coefficients a_k must satisfy certain conditions in order for the scaling function to exist and satisfy the scaling equation. There turn out to be an infinite number of sets of scaling coefficients for every even $N > 2$. It is the choice of the a_k, from among this set, which determines the detailed shape of $\varphi(x)$. A great variation exists in the possible functions φ as illustrated by the examples shown in figures 2.1 and 2.2.

Scaling functions may be selected from the class of Daubechies functions, which have several important characteristics. They are relatively smooth and have certain approximation properties (i.e., vanishing moments). These systems will be referred to as $D2$, $D3$, $D4 \ldots Dg$, where g is the *genus* of the system.

The scaling function is the basic unit from which a level of detail is constructed. This is done by considering the set of functions that can be represented as a linear combination of shifted versions of the scaling function. That is, we define a collection of functions at scale level j, which we write V_j, to be the set of functions that are linear combinations of the functions $\varphi(2^j x - k)$, where k is an integer. The factor 2^j multiplying x has the effect of shrinking the support of φ to $(N-1)/2^j$, and the shift by k moves these small functions around by a fixed fraction of their support, independent of scale. Thus a function's components at scale level j are expressed by the equation:

$$f_j(x) = \sum_{k \in \mathbf{Z}} c_{j,k} \varphi(2^j x - k) \tag{2.3}$$

where f_j is the part of f resolvable at the level j.

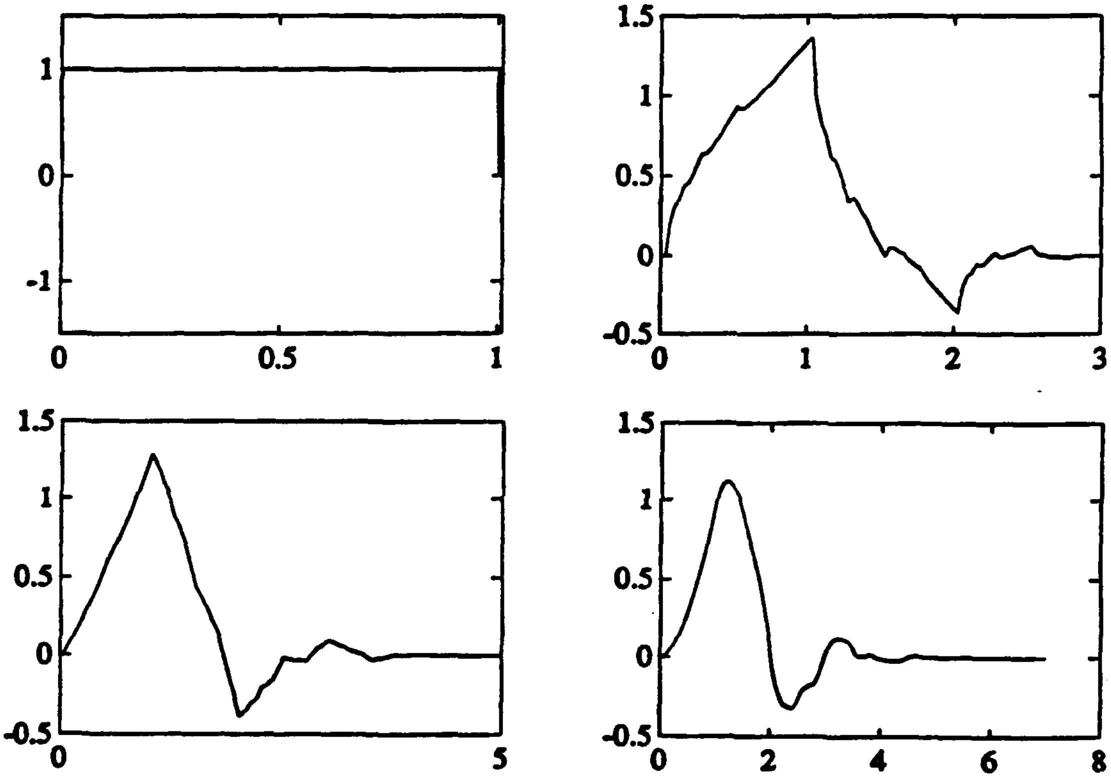

Figure 2.1: Four Examples of Scaling Functions (clockwise from upper left) "Haar", "Daubechies-2", "Daubechies-4","Daubechies-3".

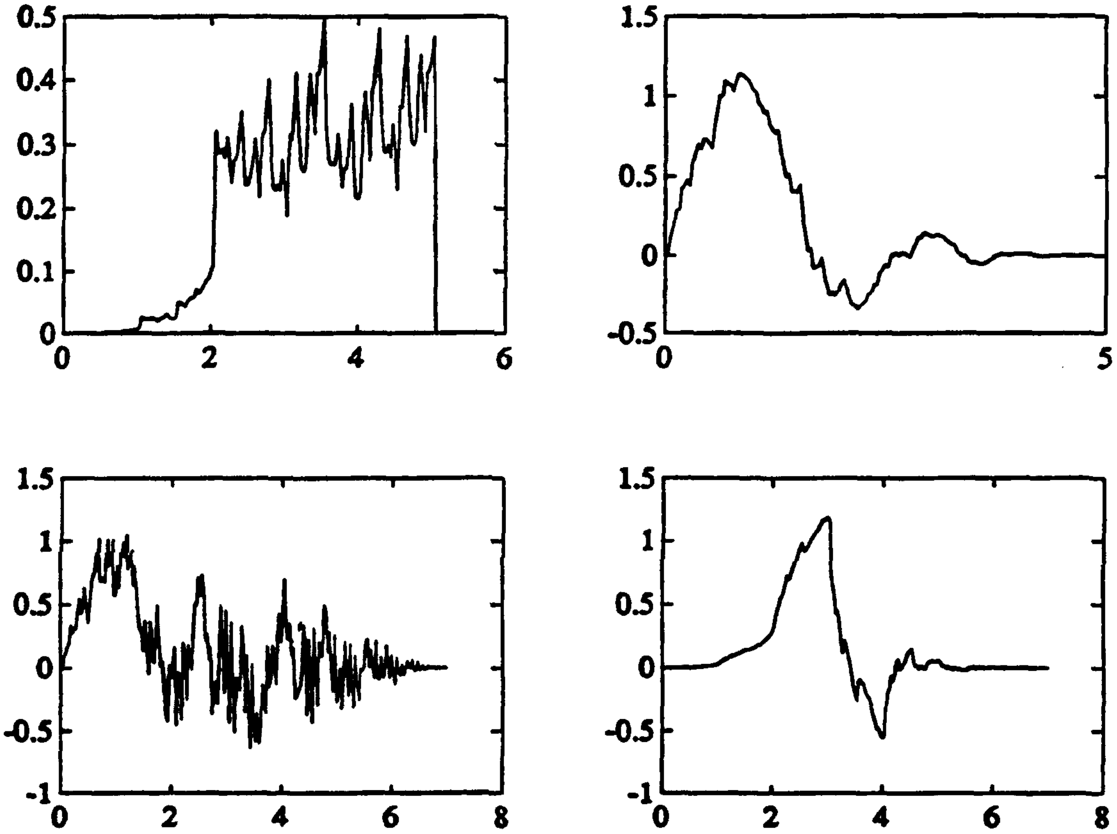

Figure 2.2: Four Assorted Scaling Functions.

This idea can also be expressed by stating that V_j is the space *spanned* by the set

$$\{\varphi(2^j x - k) \mid k \in \mathbf{Z} .\} \tag{2.4}$$

This set of functions forms an ***orthonormal basis*** for V_j. Functions in V_j are uniquely expressible as linear combinations of the basis functions, and the basis functions all have unit "energy". Thus the set of functions $\{\varphi(2^j x - k)\}$ form an orthogonal set of "templates" for the scale level V_j.

The effect of performing a transform with such a set of basis functions is to identify, within the signal, those parts or components that are similar to the basis functions at the given scale level. Similarly, a Fourier transform has oscillatory functions as a basis, and identifies the relative contribution of each frequency to the overall signal.

With shifted versions of the scaling function as a basis, the Wavelet transform will identify components that are similar to a particular shifted copy of the scaling function; that is, a representation of a function (in the scaling function basis) can identify features *locally* in time, since the scaling function is compactly supported, and locally in *scale*, because the scaling function has *structure*.

There is another, equally good, set of orthogonal "templates" for this scale level V_j. This set of "templates" gives a different type of information than the one presented above. In our previous basis, all the resolution within the scale level j was in the temporal domain: each coordinate corresponded to a position in time. The new basis will trade some of this temporal resolution for some additional structural information; it will give us two sets of coefficients, one set that represents large scale structure, while the other set represents small ("fine") scale structure. The process extracts or *filters* out the components of the scale level j that cannot be regarded as part of the coarser scale level, $j - 1$. These are taken to represent the information at scale j that is small scale compared to that part of V_j which is actually large scale information embedded in the small scale space. This operation captures the large scale features of a function f_j in the scale level $j - 1$, and retains the small scale features in a *difference space.*

2.3.2 Wavelets

This idea of dividing the scale level V_j into a coarser version of itself, V_{j-1}, and a difference space, which we will call W_{j-1}, can be compactly expressed as an *orthogonal splitting* of the space V_j into two perpendicular spaces V_{j-1} and W_{j-1}:

$$V_j = V_{j-1} \oplus W_{j-1}. \tag{2.5}$$

The reason this can be done efficiently comes from the scaling equation. Since $\varphi(x)$ can be expressed as a linear combination of translated versions of $\varphi(2x)$, the coarser scale level V_{j-1} is contained within the finer scale level V_j:

$$V_{j-1} \subset V_j \tag{2.6}$$

Repetition of this argument shows that

$$V_{j-1} \subset V_j \subset V_{j+1} \subset V_{j+2} \dots \tag{2.7}$$

The difference space, W_{j-1}, contains all that remains when the coarser scale information is removed. However, since W_{j-1} is contained in V_j, it is also expressible as a linear combination of translates of $\varphi(2^j x)$.

The actual set of functions that are used to *span* the space W_{j-1} are the orthonormal basis formed by the functions

$$\psi(2^j x) = \sum_{k=0}^{N-1} (-1)^k a_{N-k-1} \varphi(2^{j+1} x - k) \tag{2.8}$$

or in the case of $j = 0$,

$$\psi(x) = \sum_{k=0}^{N-1} (-1)^k a_{N-k-1} \varphi(2x - k). \tag{2.9}$$

Notice that the signs now alternate in the sum, and the order of the coefficients a_k has been reversed ($k \rightarrow N - k - 1$). These changes make $\psi(x)$ orthogonal to $\varphi(x)$. The full basis for W_j is formed by taking shifted versions of ψ, i.e., **Basis**$(W_j) = \{2^{j/2}\psi(2^j x - k) \mid k \text{ an integer }\}$. The support of $\psi(x)$ is easily seen to be the same as the support of $\varphi(x)$, and this is true for the shrunken versions as well, that is, the support of $\psi(2^j x - k)$ is the same as the support of $\varphi(2^j x - k)$. The normalization term $2^{j/2}$ maintains unit energy

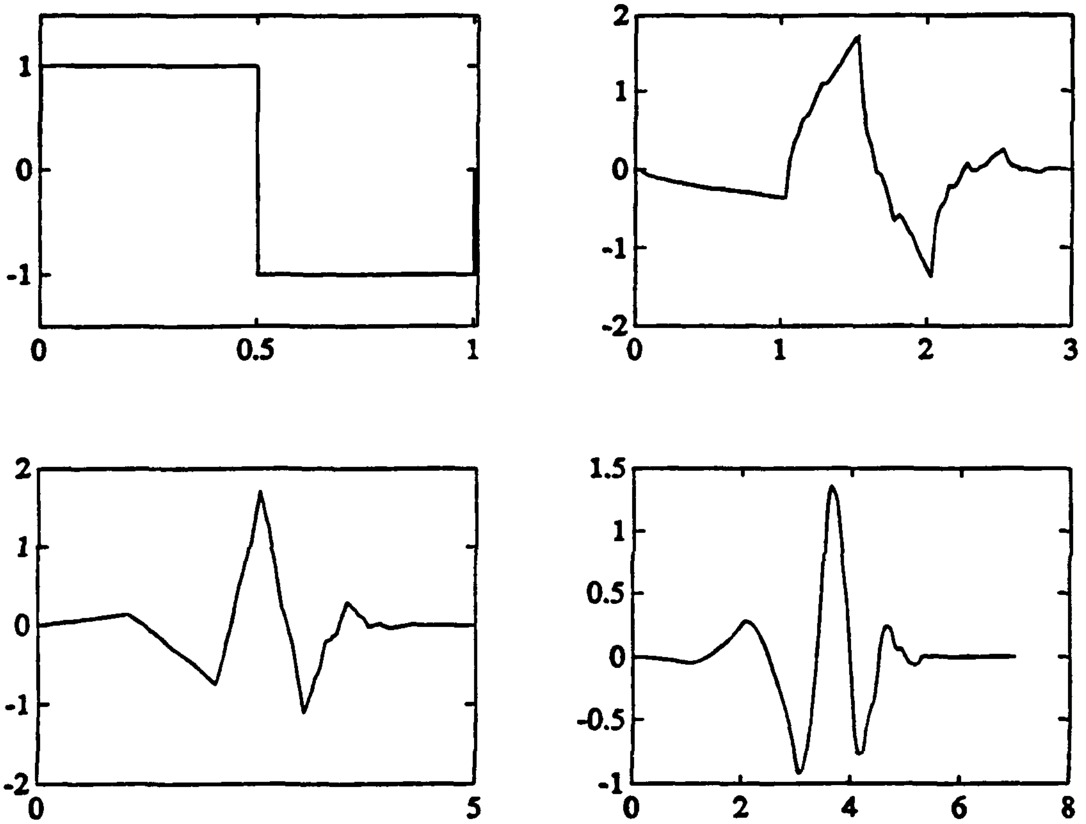

Figure 2.3: Four Examples of Wavelets (clockwise from upper left) "Haar", "Daubechies-2", "Daubechies-4","Daubechies-3".

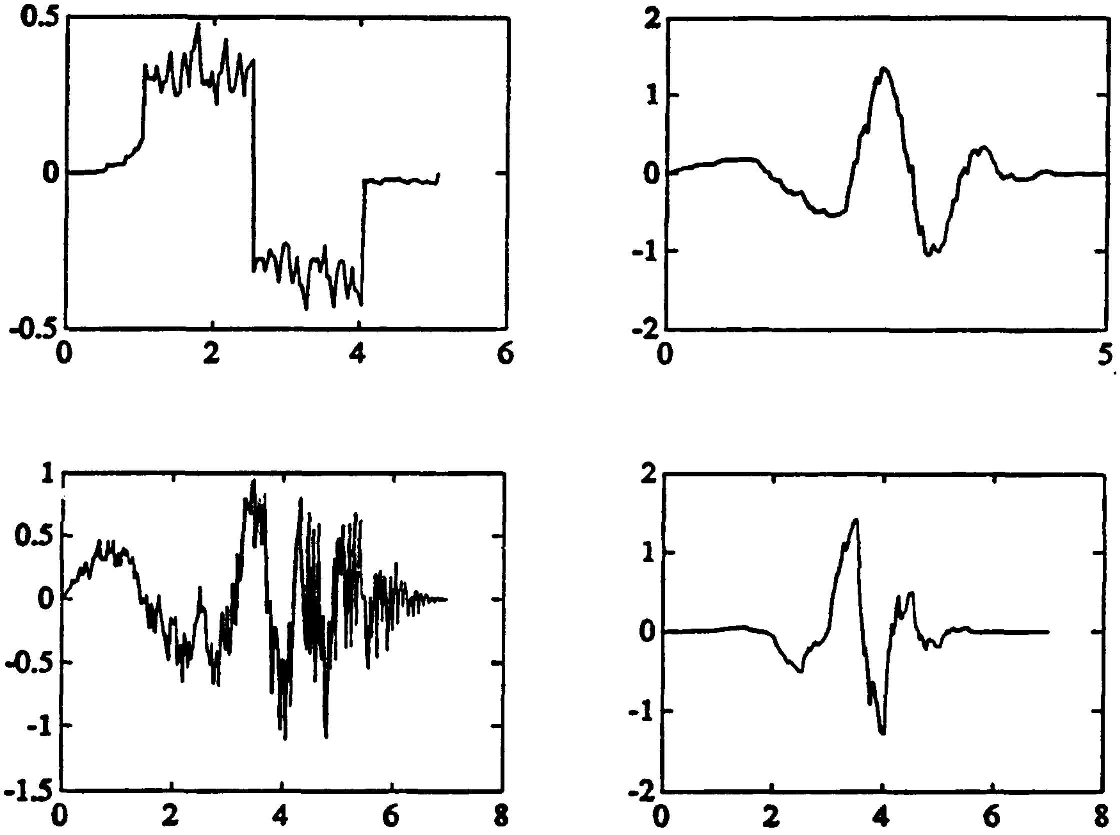

Figure 2.4: Four Assorted Wavelets corresponding to the Scaling Functions in Figure 2.2.

in the functions. Figures 2.3 and 2.4 are the wavelets that correspond to the scaling functions presented in figures 2.1 and 2.2.

Thus the transformation from the first representation of V_j, where f_j was expressed as a linear combination of shifted versions of $\varphi(2^j x)$, to the new representation, where f_j is expressed as a sum of translates of $\varphi(2^{j-1}x)$ *and* translates of $\psi(2^{j-1}x)$, gives us new information about the shapes and structures, perhaps frequencies, present in f_j. This is at the expense of some temporal resolution, because the new basis functions are twice as long. The new basis incorporates the inter-relationships among larger subsets of the data, providing correlative information. As a result of the spectral refinement, there has been a loss of temporal resolution.

The scaling function, being the basis for all the spaces V_j, forms the connection between these spaces via the scaling equation. The basic wavelet, ψ, represents the differences between the scale levels.

2.3.3 The Wavelet Transform

The exchange of temporal resolution in V_j for scale resolution in the division of V_j into V_{j-1} and W_{j-1} forms the basic unit of the Wavelet transform. Since the definition is independent of scale, it can be repeatedly applied in the same way. Furthermore, the operations involved in the transformation require only the expansion coefficients of the function $f(x)$ in the basis at the current scale, and not the actual values of the function. The computation is very *simple* and *efficient* because of the close link between the functions φ and ψ.

The basic operation involved in a Wavelet transform is the conversion of temporal resolution into structural or spectral information. This basic single step, essentially a filter, exchanges half the temporal resolution of a signal for twice the frequency resolution; the product of the two remains the same[6]. More importantly, this operation can be repeated to gain any desired level of detail in the structural or spectral realm, while only imposing a reciprocal loss of resolution in the temporal domain. This is in sharp contrast to Fourier transform techniques, where one either gets all the available frequency information, or none of it, with no intermediate stages of knowledge available.

[6] This is a case of the Heisenberg Uncertainty Principle.

The wavelet transform allows one to move gradually between the two extremes present in the Fourier transform, successively gaining shape or structure information. In a sense, the wavelet transform interpolates between the frequency, or structure, domain and the temporal domain. This step by step transformation can be understood in terms of trade-offs between relative time resolution and relative frequency resolution, but its features are most simply understood by viewing them in *phase space*.

Phase space is a two dimensional plane in which we place frequency (or more generally, some distinguishing structural property) on the y-axis and time on the x-axis. Any time or frequency based transform will be characterized by its division or partition of phase space into "resolution cells". Each resolution cell corresponds to a single basis function, and thus represents a specific occurrence. The shape of the resolution cell reflects the characteristics of the basis function, its width in time representing the support or extent of the basis function, and the width in frequency representing its bandwidth.

If we consider the original signal, we know its values at each point in time while we know nothing about its frequency characteristics. The resolution cells are vertical strips, each representing the fine temporal resolution and the complete lack of frequency information. When the signal is Fourier transformed, exactly the opposite happens. Now the frequency content of the signal is known perfectly, but all knowledge of location in time has been lost. Now the resolution cells are horizontal strips of width equal to $1/2N$, each reflecting the magnitude of a single frequency, while carrying no information about when that frequency was present.

In general, the area of a resolution cell cannot be decreased, but its shape may change. If we assume that these cells remain rectangular, then a cell may be made narrow in one direction, but only if it is simultaneously widened in the other direction. This is exactly what happens when one applies the basic wavelet scale separation operation. One exchanges a factor of two in the width of the resolution cells for a factor of one half in their height. This operation is displayed in figure 2.5. The wavelet transform provides a "zoom-lens" approach to signal representation. Any degree of structural information can be acquired, but only at the expense of enlarging the "field of view" temporally. Conversely, one may zoom in on a particular temporal scale, but only at the expense of global structural information.

Since the transform is defined in terms of operations on the *coefficients* of the representation, and *not* the actual values of the scaling or wavelet

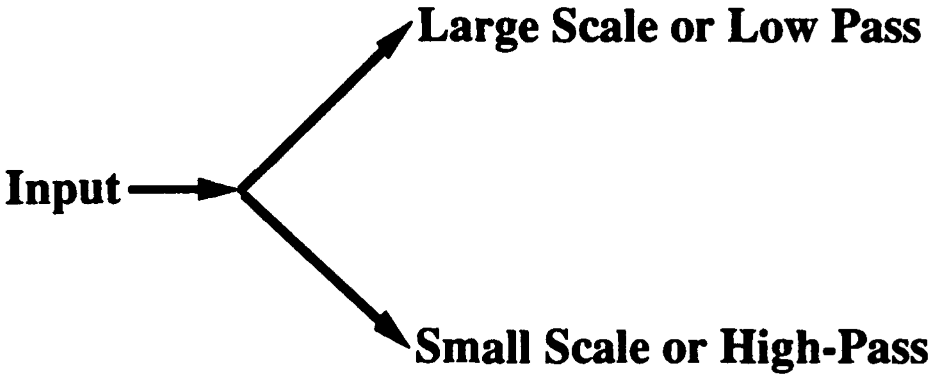

Figure 2.5: Conceptual Scale Separation Operation of Wavelet Transform.

functions, the output from a single stage of the transform is exactly what the next stage requires for input. This easily pipelined, recursive structure is what makes the wavelet transform rapidly computable. While many such structures are made possible by the wavelet transform, one in particular, the one-sided or Mallat transform, has proven to be exceptionally useful in analyzing signals. Figure 2.6 is a conceptual diagram of a 3 level, one-sided wavelet transform. The large scale (φ) output coefficients from the first stage are the input for the second stage, etc. The number of data points at each level is reduced by a factor of two by the conversion of temporal information into spectral information. Figure 2.7 illustrates a four level, one-sided wavelet transform. We have included plots of the Daubechies-5 basis functions for this transformation. Figure 2.8 is a plot of the spectral characteristics of the wavelet transform shown in figure 2.7. Notice the broad high frequency response characteristics of the smallest scale wavelets which corresponds to their fine temporal resolution. The large scale wavelets have just the opposite characteristics, sharp spectral response and coarse temporal resolution.

The method outlined for the one-sided transform method can be generalized to include any "binary tree" structure desired. The basic transformation unit is repeatedly applied, taking the outputs of one level as the inputs for the next level.

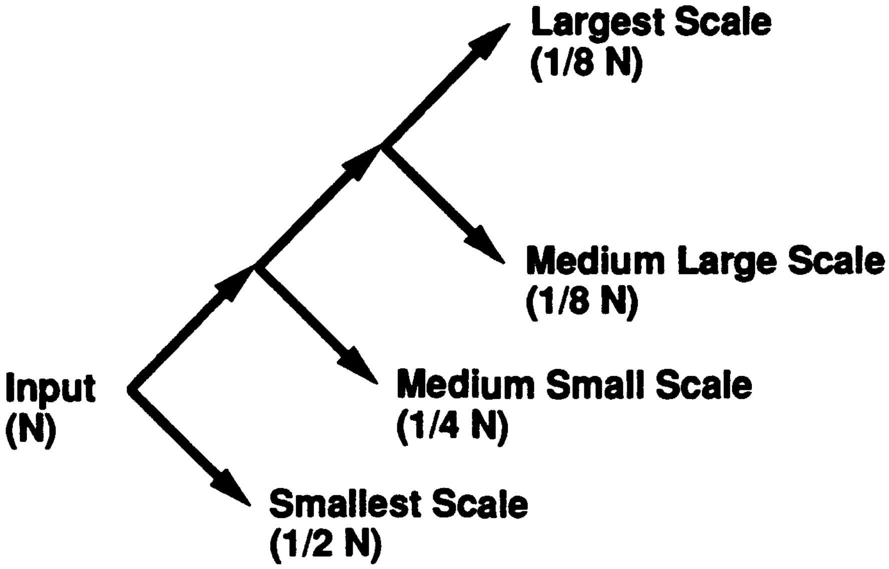

Figure 2.6: Conceptual 3 Level, One-sided Wavelet Transform.

2.4 Wavelet Signal Processing Methods

Wavelet signal processing methods use the properties of wavelet transforms to extract signal information about events, features, modes and other characteristic behaviors. There is no single wavelet transform of a signal, but a family of them, all equivalent to the original signal in information content.

The choice of wavelet technique will depend on the specific signal processing requirements. Transient detection and harmonic analysis often call for octave band decomposition that extracts the harmonic signatures of the signals, preserves fine temporal resolution and is computationally very efficient. The choice of wavelet basis functions can have a significant effect on the ability to separate signals of a particular type from noise or other interference. The technique is flexible in that additional spectral resolution can be obtained if required.

A slight modification of this method can be applied to remove noise from signals. Signal features that exist on a small number of scales can be isolated by performing a wavelet transform and discarding coefficients at all the scales except the "target" scales and then inverting the transform to reconstruct

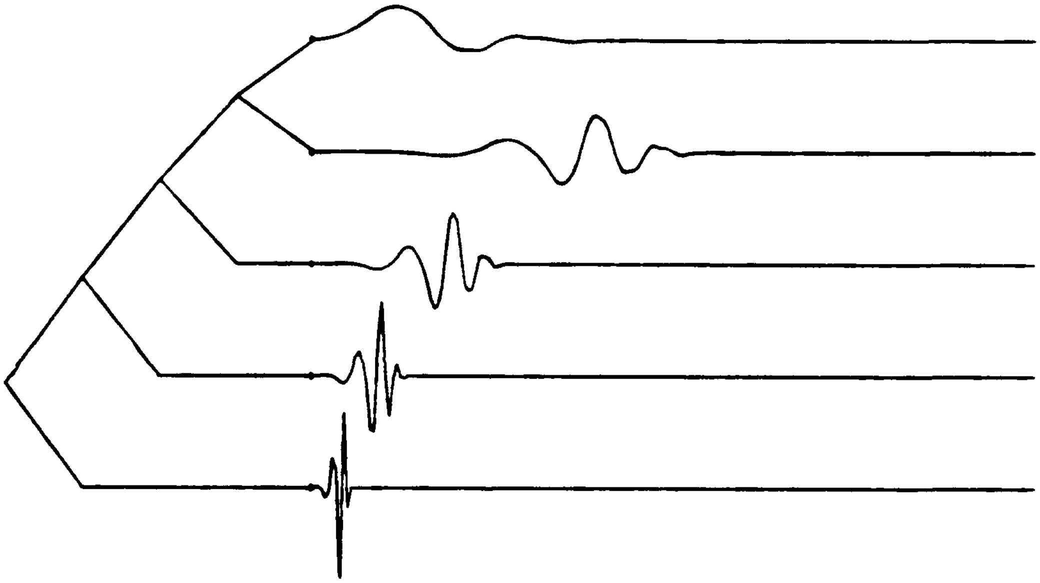

Figure 2.7: Four Level, One-sided Wavelet Transform with Characteristic Functions for the case of Daubechies-5.

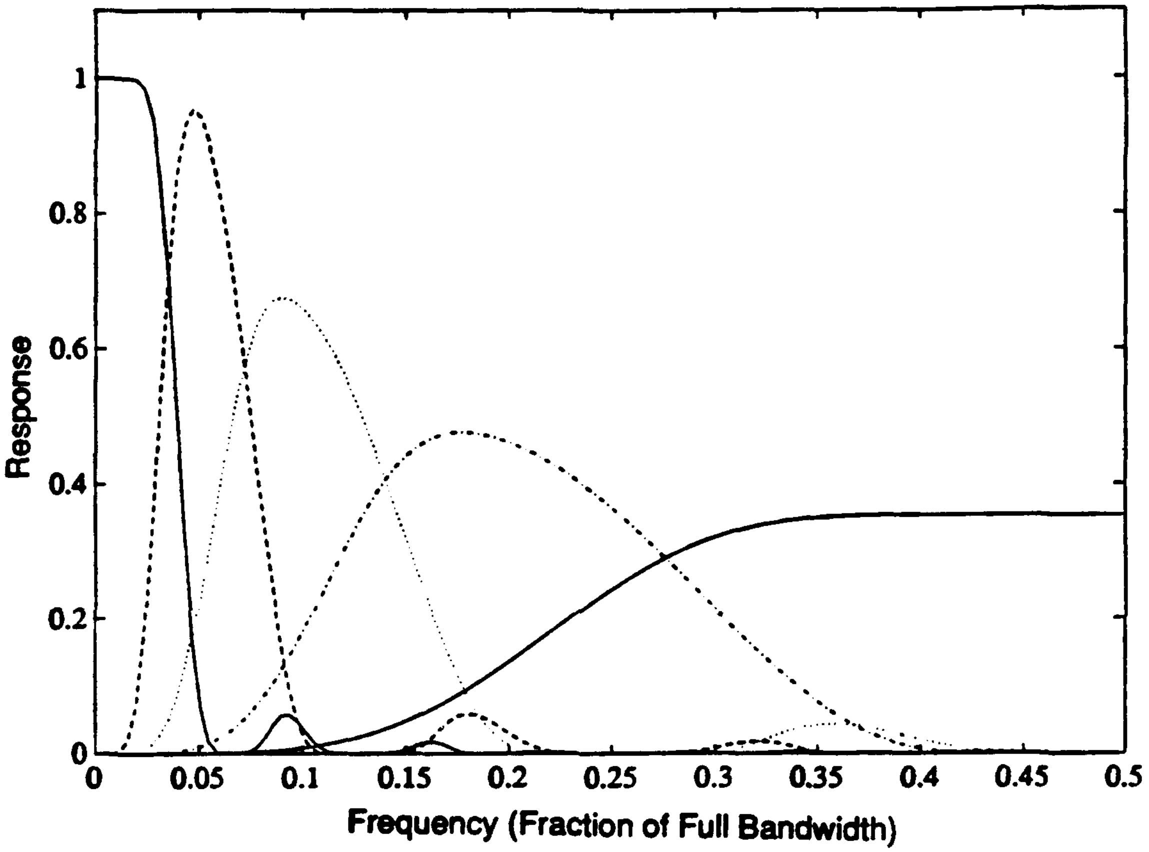

Figure 2.8: Spectral Characteristics for the Four Level, One-sided Wavelet Transform in figure 2.7. Note the narrow low-frequency response of the long scaling function and the broad frequency response of the shortest wavelet.

the signal with the noise removed. The adaptive selection of coefficients is possible based on fixed or floating thresholds.

The wavelet transform is lossless at each level and forms a hierarchy of successively finer spectral refinements. The signal energy at any node in a wavelet filter bank can be further resolved in frequency, but will never migrate into an adjacent sub-band. This allows the signal energy in each node to be evaluated in real time and guide the development of the filter bank structure. Low energy nodes will only generate sub-nodes with even less energy. The ability to "prune" the processing tree significantly reduces the computational cost of the transform method.

The structure and computational complexity of several wavelet signal processing methods are discussed in the next chapter.

2.5 The Computational Complexity of Wavelet Transforms

This section is concerned with the computational efficiency of wavelet-based analysis techniques. The computational complexity of several types of wavelet transforms is developed and comparisons are made to the computational complexity of the FFT. The main results are:

- Evaluation of wavelets and scaling functions has $O(n)$ computational complexity.

- Wavelet calculations are parallelizable and can be pipelined. Wavelet transforms can be calculated in $O(n)$ operations and in $O(\log n)$ time if concurrent computation is employed.

- Wavelet algorithms depend parametrically on the scaling coefficients, so algorithms can be embodied in programs that are structurally independent of the system of scaling coefficients.

- Wavelet stream processing techniques can run in real time at full resolution. The calculation density, i.e., the number of operations per datum, does not grow logarithmically, as it does with an FFT, or require artificial framing, segmenting or windowing to render real-time operation feasible.

- Certain classes of wavelet coefficient systems can be exactly computed in a digital computer without roundoff error.

We will discuss both block and stream processing applications. Comparisons with the FFT are natural here, since the FFT is a commonly used, well understood, and highly optimized algorithm. It is a convenient benchmark for comparison. We ignore pre- and post-processing requirements assuming that the output from the wavelet transform is the desired result.

For the case of block processing, the input block size is the factor that determines the computational cost. The first case is a wavelet transform that resolves only a portion $R = 1/2^J$ of the total bandwidth, and resolves it as finely as possible using recursive wavelet techniques. The simplest example of such a transform is the familiar Mallat Transform, which calculates the decomposition of a signal on the basis of scale, i.e., it "homes in" on low frequencies. This fundamental structure requires

$$OPS(Mallat) = \alpha(K)N(1 - 1/2^J) \text{ \# of operations} \tag{2.10}$$

where $\alpha(K) = (K + 1)$ multiplies and (K) additions per output point. The number of input data is N, and J is the finest level calculated; that is, the basic wavelet decomposition operation is applied J times. The wavelet transform can be implemented as a finite impulse response (FIR) filter with the number of "taps" equal to the number of nonzero wavelet coefficients. The number of nonzero coefficients is K, which we have also called the length of the wavelet coefficient matrix. Note that whatever the depth of the decomposition, the operation count never exceeds $\alpha(K)N$. This operation count also applies to *any* wavelet transform that "zooms in" on a single location in phase space, allowing other side-bands to remain unchanged. These are not *partial* wavelet transforms, each is a complete representation in a wavelet basis. Each is, however, a partial frequency decomposition, but that is a powerful advantage because **only the minimum amount of computation is performed**.

In applications where one wishes to resolve frequencies (or some other structure) to some pre-specified resolution (say $1/2^J$), and one is interested in a small number of sub-bands, *wavelets are very computationally efficient.* This case includes the Mallat Transform, and any other wavelet processing scheme that generates only a small subset of the finest resolution cells.

For the case of *stream* processing, the comparison with the FFT is no longer appropriate. An FFT simply cannot provide the pipelined operation of a wavelet transform. Even a windowed or Short Time Fourier Transform can only resolve successive segments or frames of the data, and within each frame, it must perform a full FFT with the corresponding lack of flexibility in the resolution. On the other hand, for any fixed filter structure (fixed tree) wavelet transform technique that is designed to generate a specific decomposition in frequency which has a finest resolution level J, the entire transform can be implemented as a pipeline, and **always has** $O(n)$ **complexity**. The coefficient multiplying N in the number of operations required is determined by how many subbands are expanded and the particular level of detail. The complexity is bounded above by $O(NJ)$, and therefore for a stream processing application, represents a constant processing delay for the calculation of a full wavelet transform.

Chapter 3

Wavelet Transient Signal Processing

This chapter begins with a description of the transient signals used for this study. This is followed by a description of the signal processing algorithms used for signal detection, feature extraction and classification. The final section summarizes the results of the signal processing experiments that were conducted during the course of this project.

3.1 Transient Signal Data

The transient signal data used in this consisted of twenty-four records of digitally sampled signals. These records contained the transient of interest as well as interference. The interference was a mixture of white Gaussian noise, quantization noise, power supply/instrumentation noise and bursts of nearly constant frequency sinusoids. The sampling frequency was well above the Nyquist rate thus preserving fine signal structure.

Figure 3.1 is a typical signal record. It contains strong interference components in the first half of the record. The transient signal of interest is located slightly after (to the right of) the center. The final third of the record contains only noise. Figures 3.2 through 3.25 are plots of the signals considered in this study.

The signals are numbered 1 through 24. They are also identified by a source file identifier.

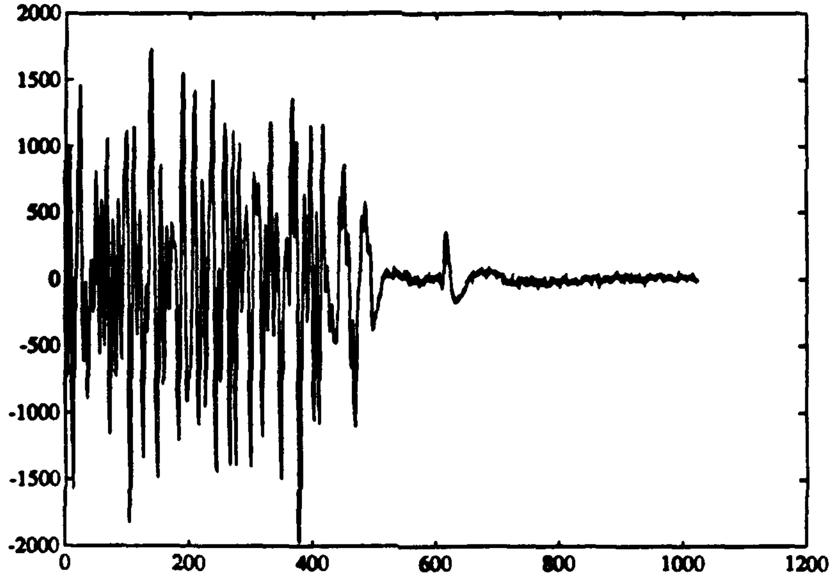

Figure 3.1: Typical signal and environment.

The signals were grouped into seven "truth" groups based on apriori knowledge of the source of each signal. Figures 3.26 through 3.32 are stacked plots of the transient signals (as segmented by the detection process) in each group. All the signal groups seem reasonable except that signal number 10 in group 2 seems to be mis-assigned.

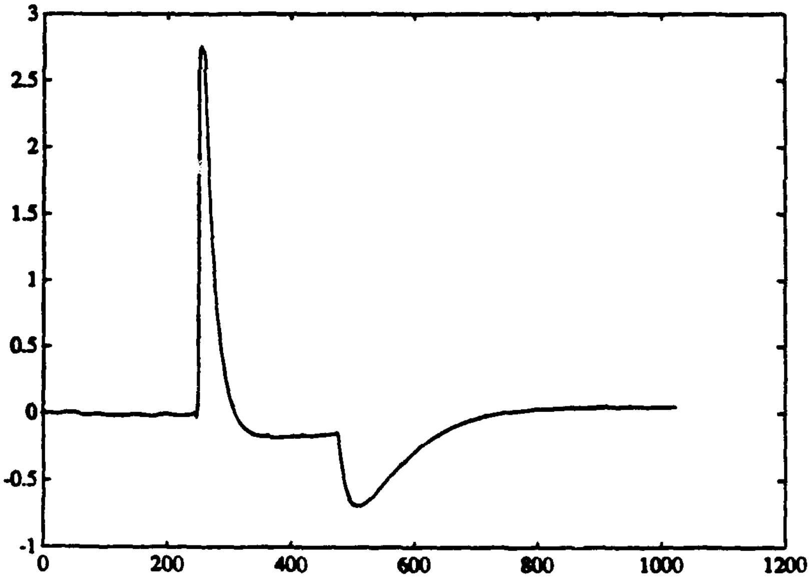

Figure 3.2: Signal number 1, (mmm1csa).

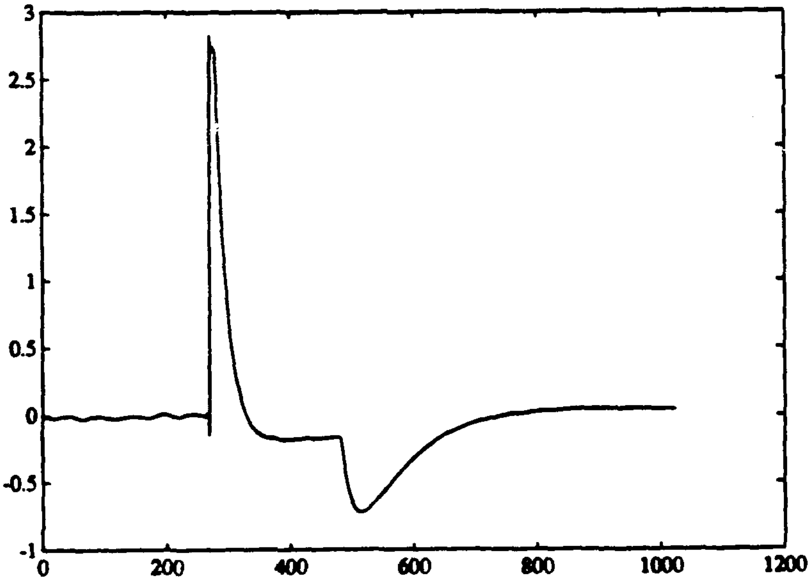

Figure 3.3: Signal number 2, (mmm2csa).

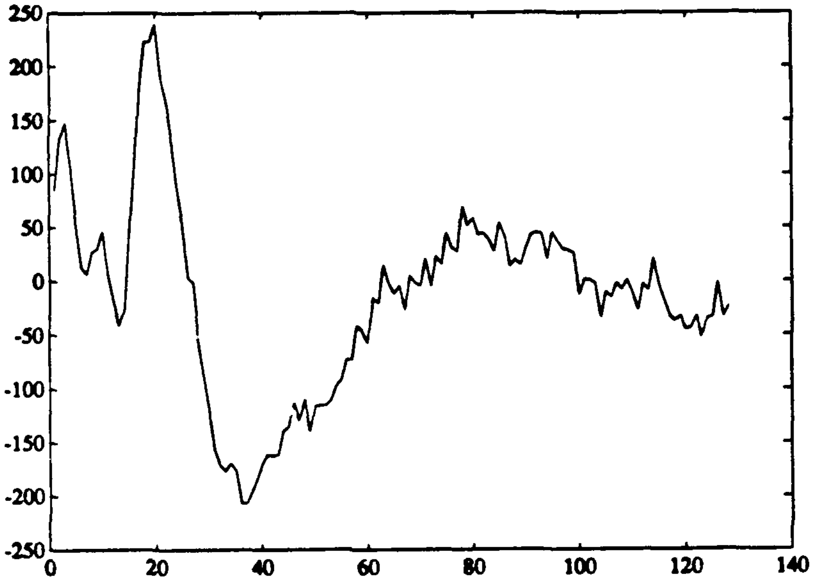

Figure 3.4: Signal number 3, (mmmlmmmb).

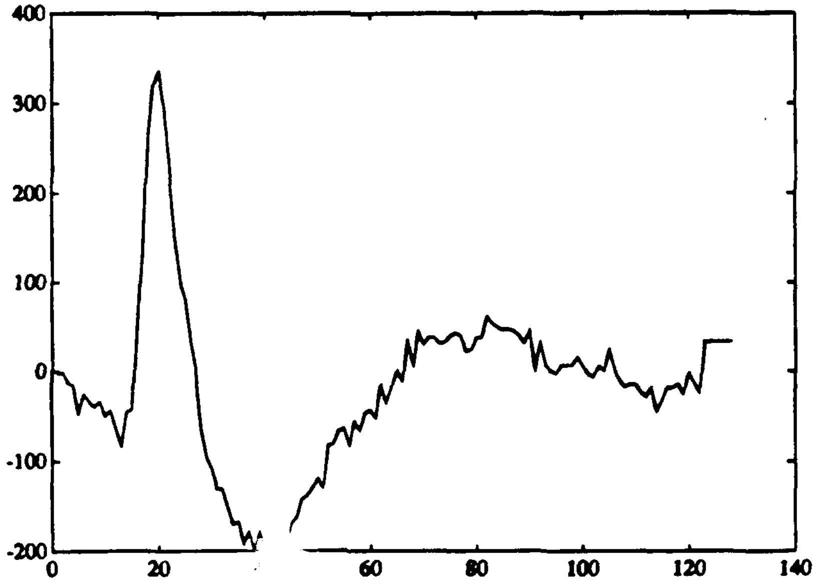

Figure 3.5: Signal number 4, (mmmlmmmd).

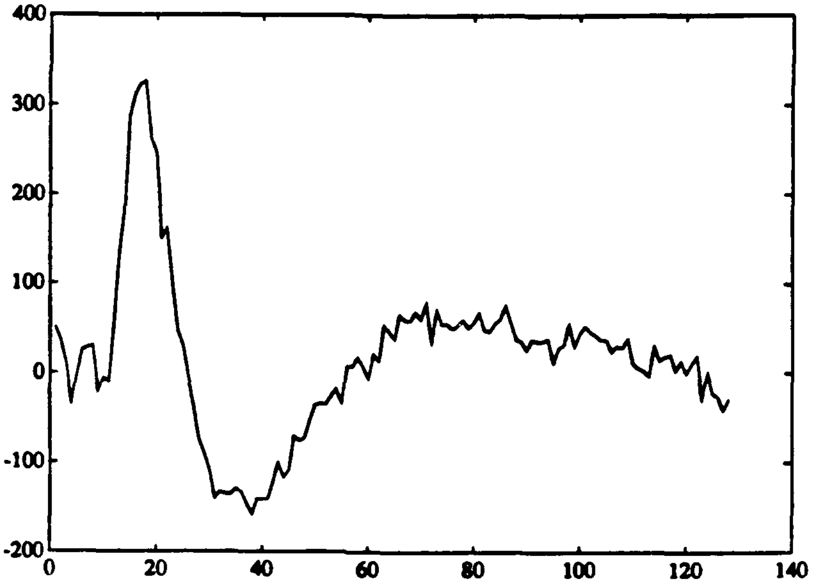

Figure 3.6: Signal number 5, (mmmlmmmf).

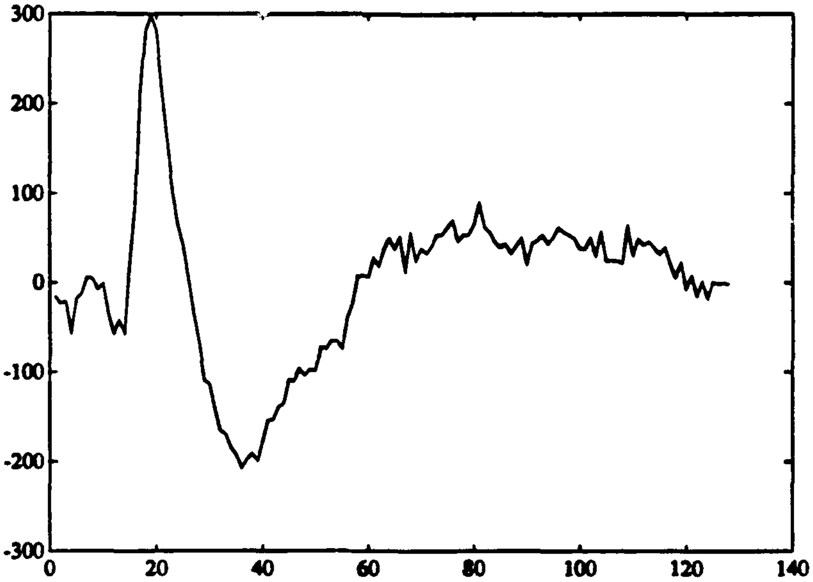

Figure 3.7: Signal number 6, (mmmlmmmg).

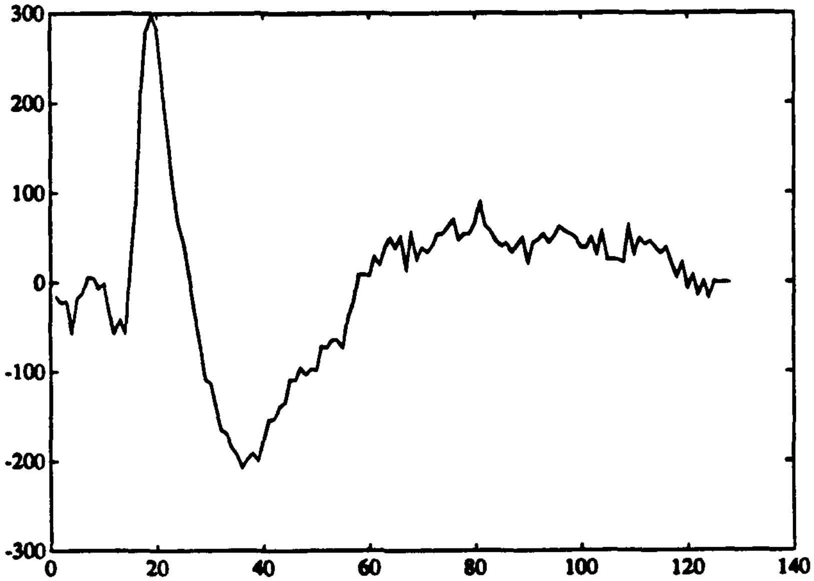

Figure 3.8: Signal number 7, (mmm2mmma).

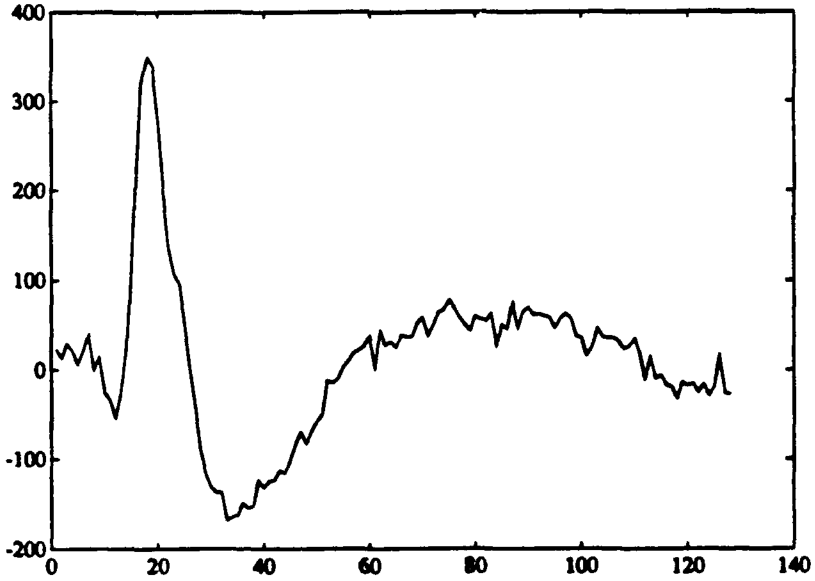

Figure 3.9: Signal number 8, (mmm2mmmd).

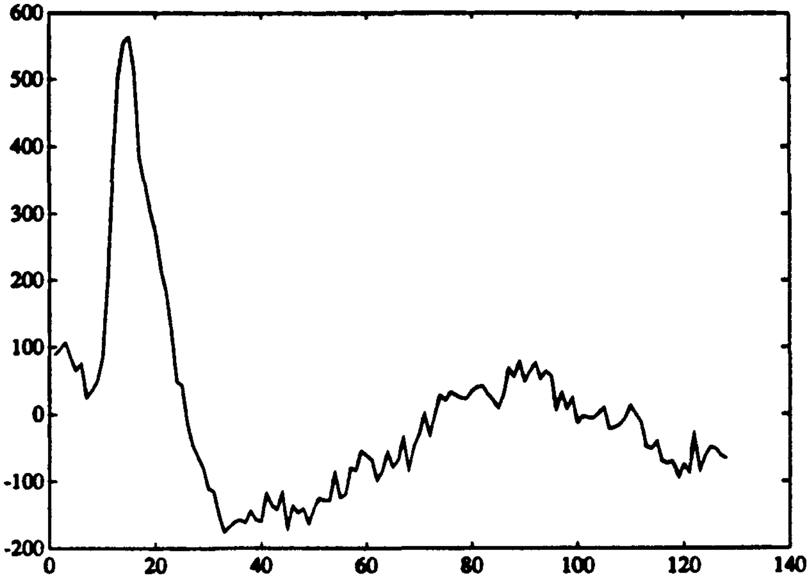

Figure 3.10: Signal number 9, (mmm3mmmb).

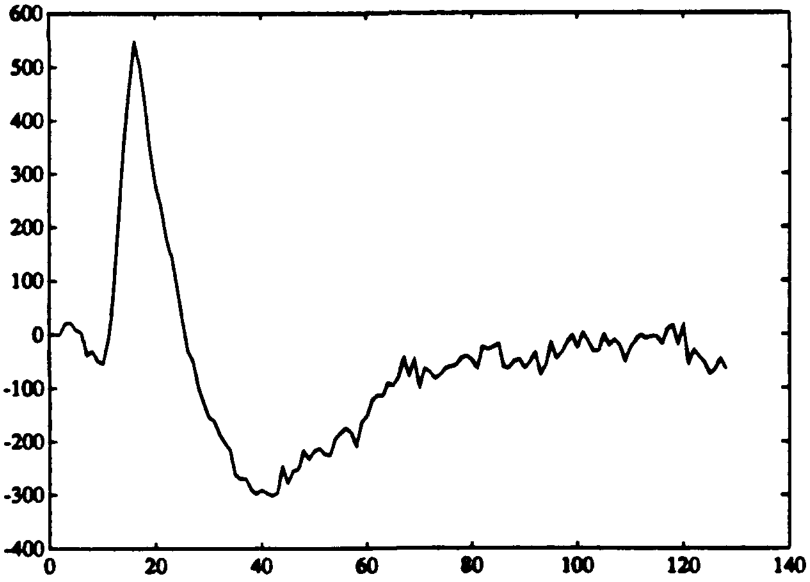

Figure 3.11: Signal number 10, (mmm3mmmd).

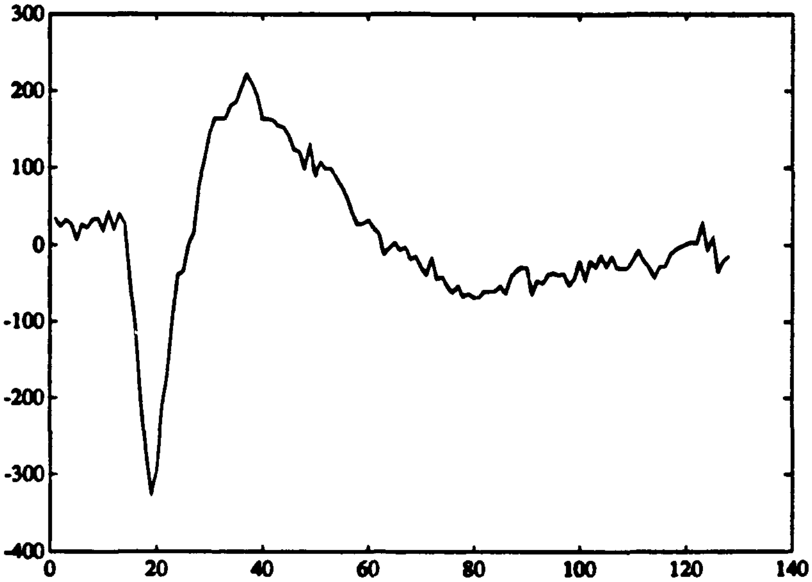

Figure 3.12: Signal number 11, (mmmlmmmc).

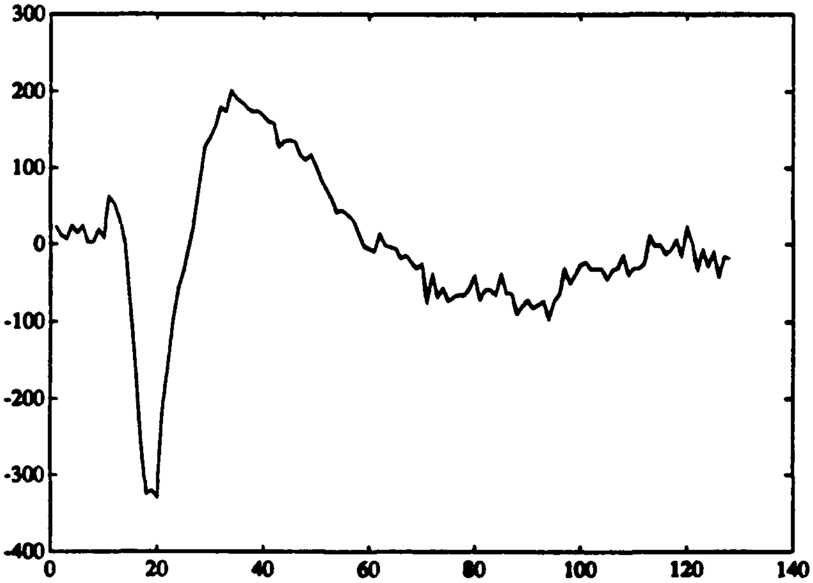

Figure 3.13: Signal number 12, (mmmlmmme).

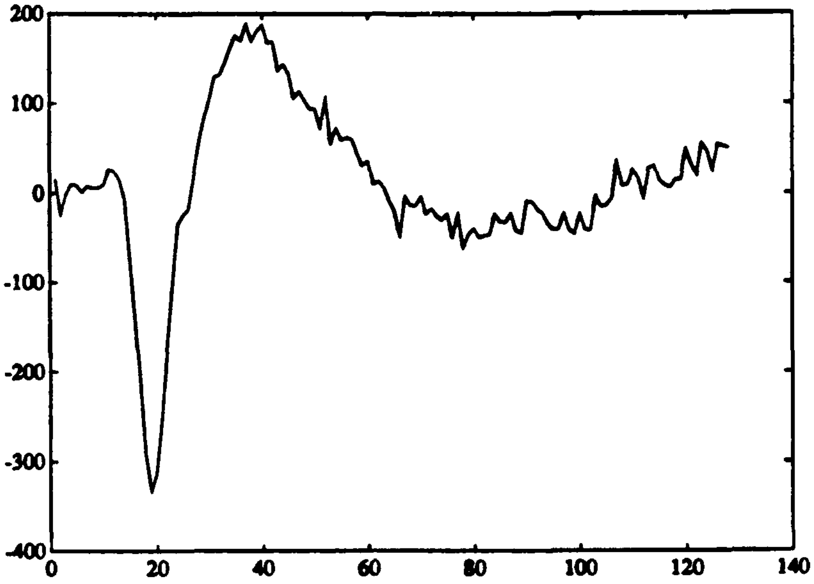

Figure 3.14: Signal number 13, (mmm2mmmb).

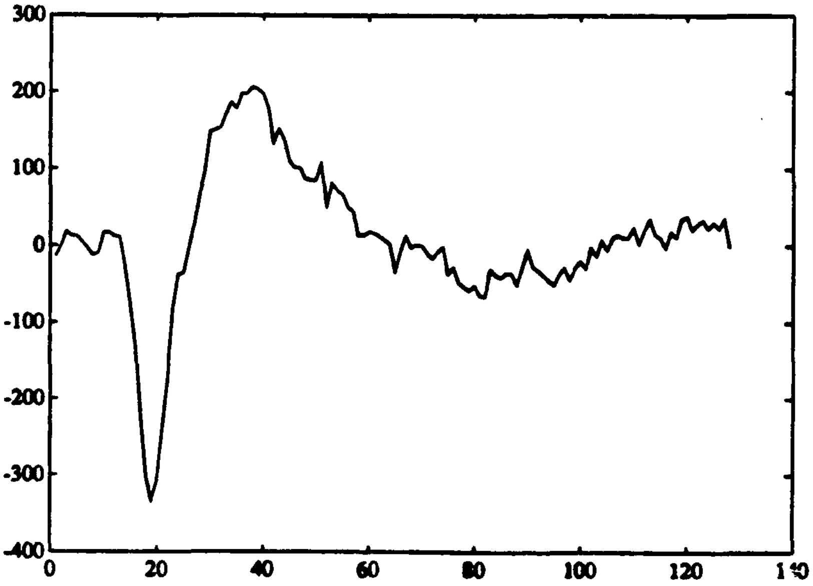

Figure 3.15: Signal number 14, (mmm2mmmc).

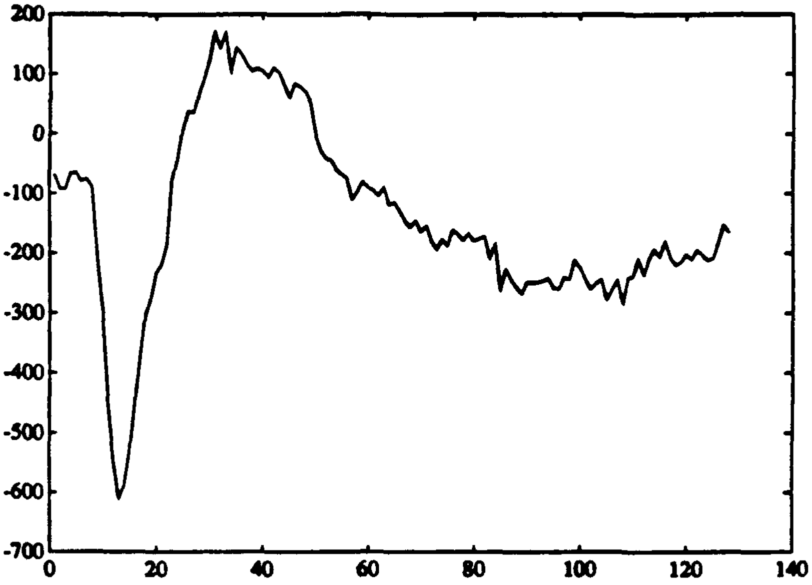

Figure 3.16: Signal number 15, (mmm3mmma).

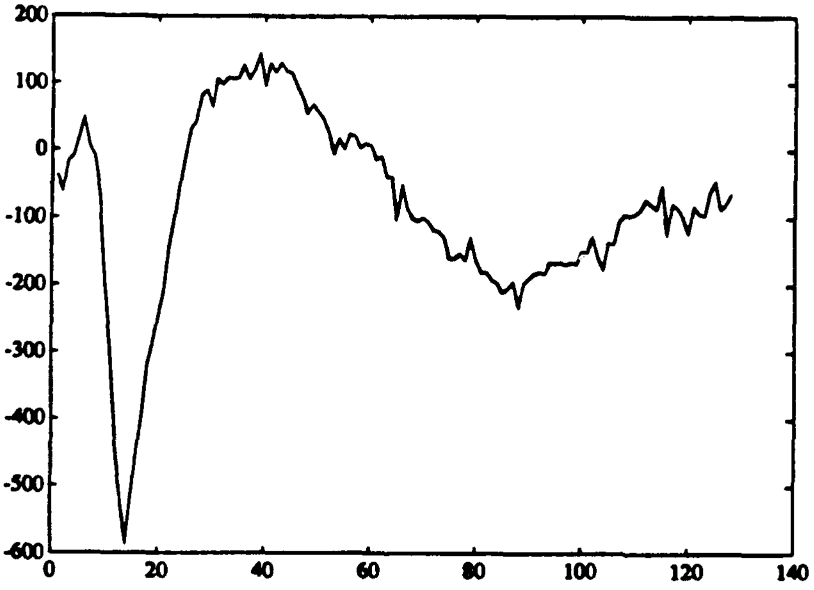

Figure 3.17: Signal number 16, (mmm3mmmc).

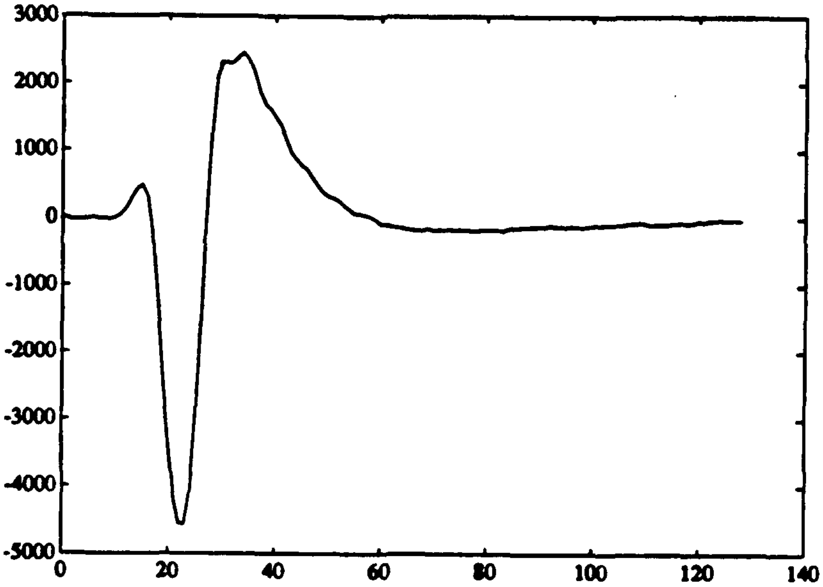

Figure 3.18: Signal number 17, (mmm1mmmh).

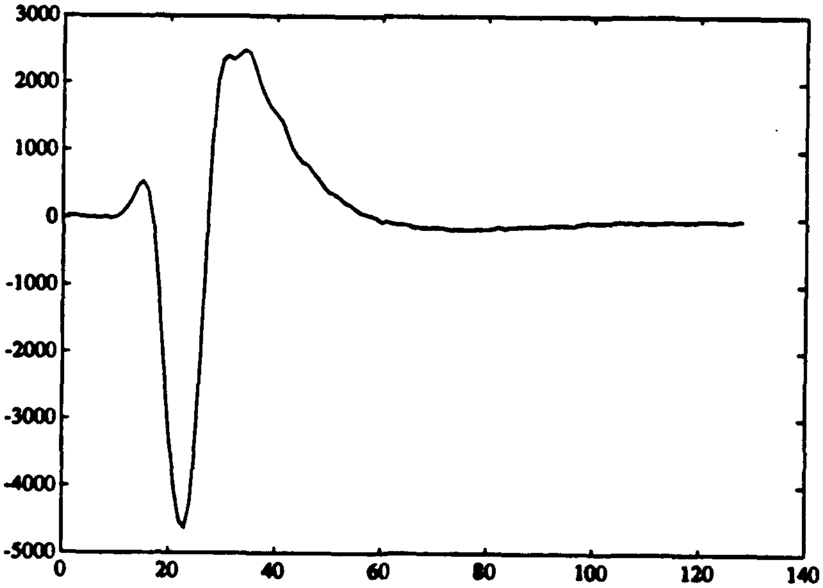

Figure 3.19: Signal number 18, (mmm2mmme).

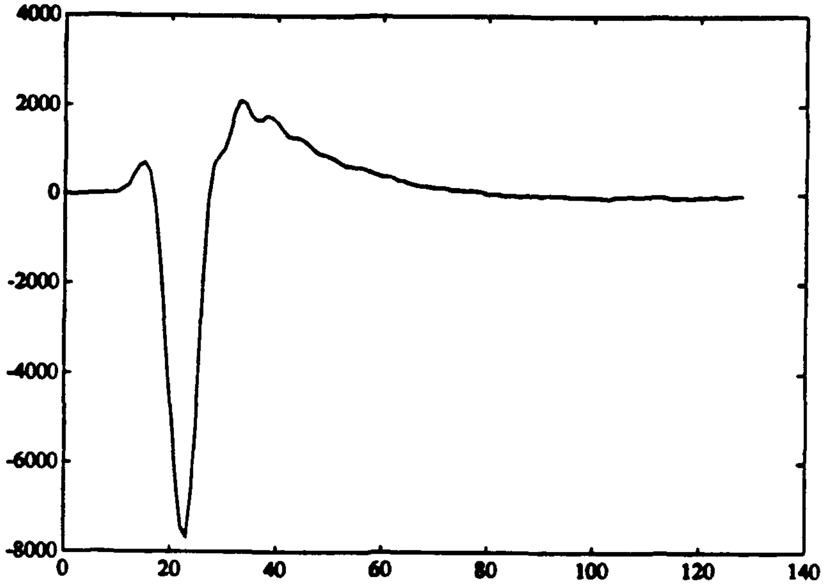

Figure 3.20: Signal number 19, (mmm3mmme).

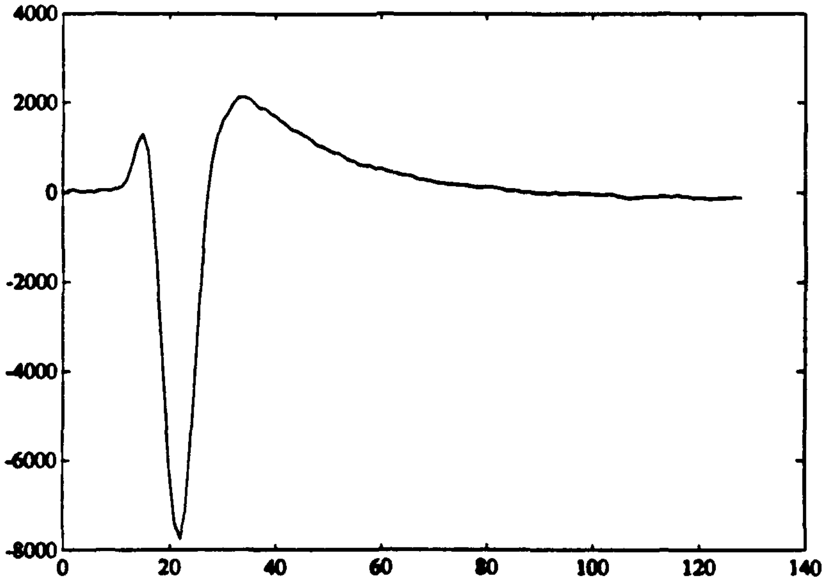

Figure 3.21: Signal number 20, (bs1bsa).

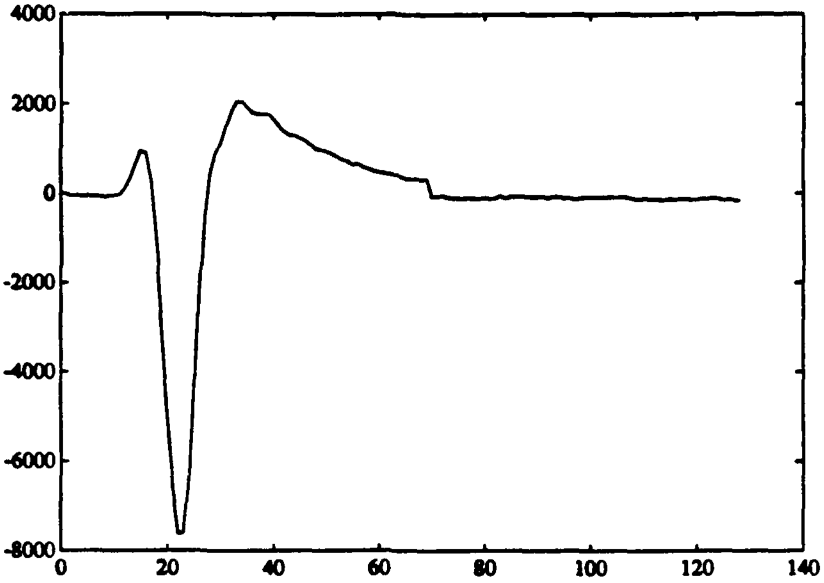

Figure 3.22: Signal number 21, (bs2bsa).

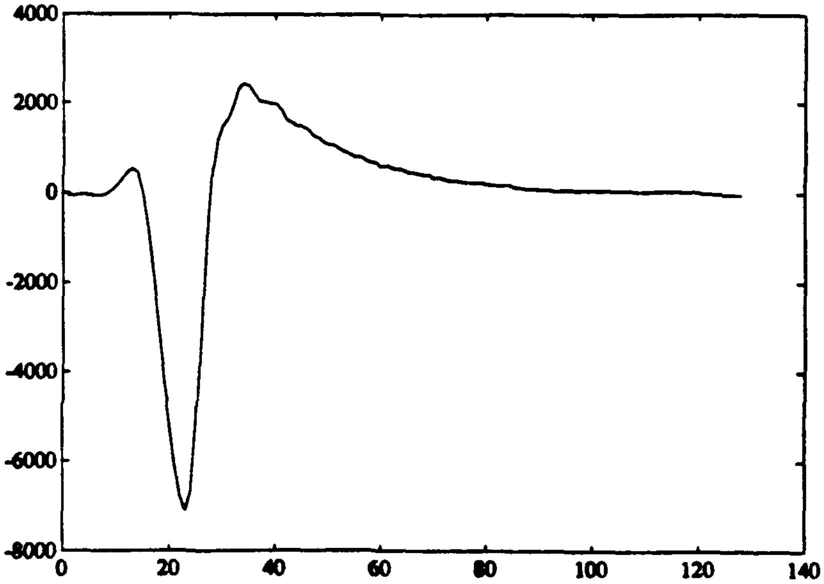

Figure 3.23: Signal number 22, (saw1sawa).

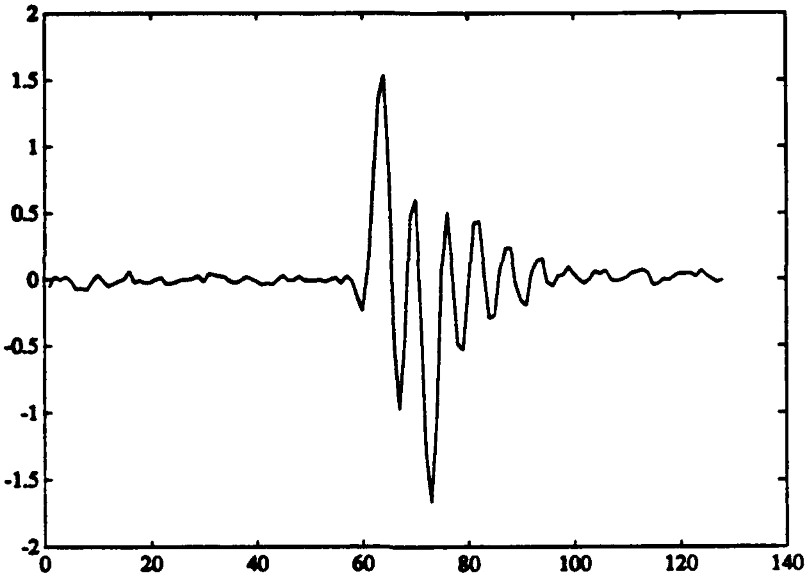

Figure 3.24: Signal number 23, (t6aaa).

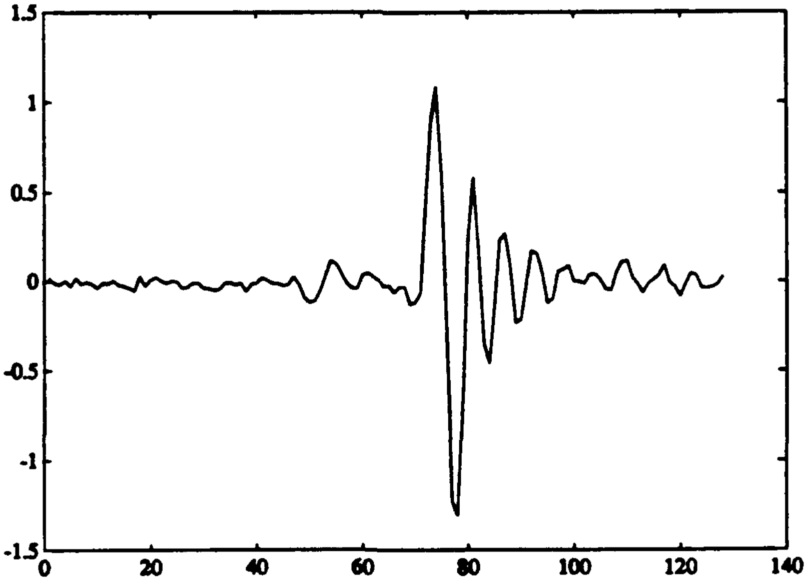

Figure 3.25: Signal number 24, (t14aaa).

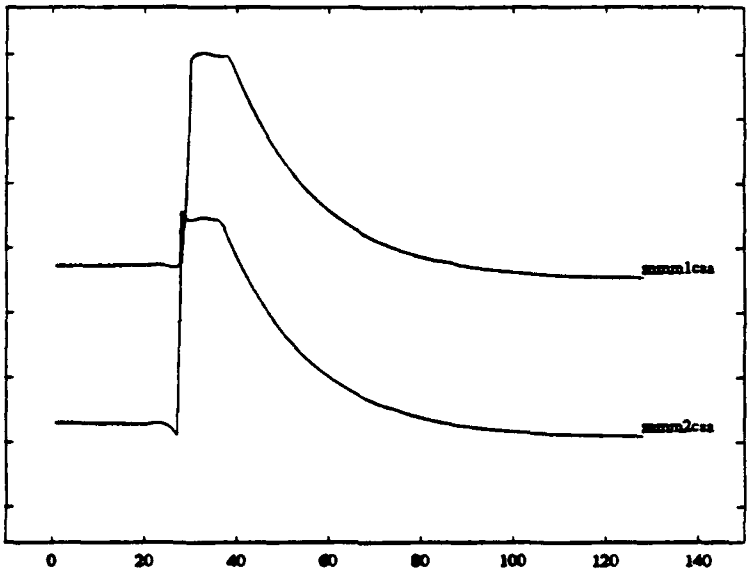

Figure 3.26: Group 1, (Signals 1-2).

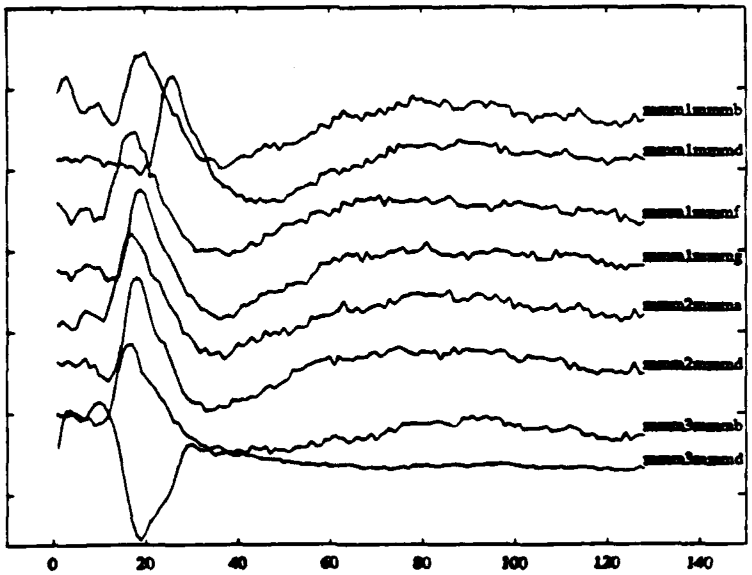

Figure 3.27: Group 2, (Signals 3-10).

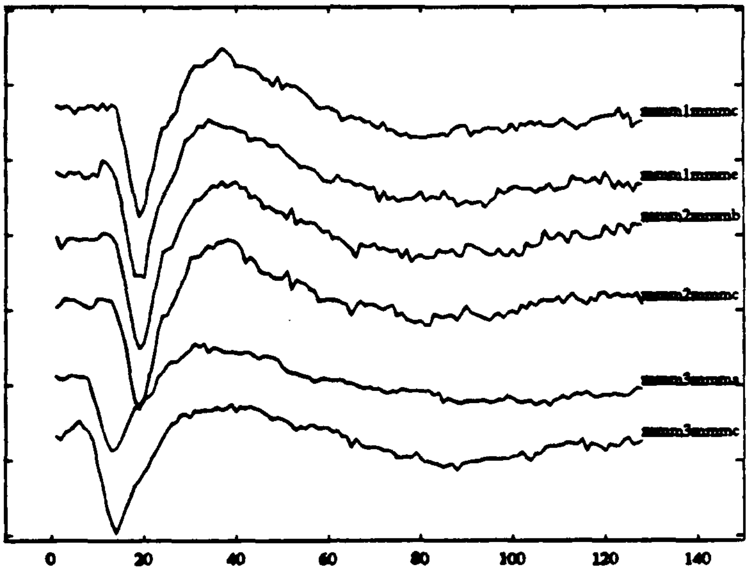

Figure 3.28: Group 3, (Signals 11-16).

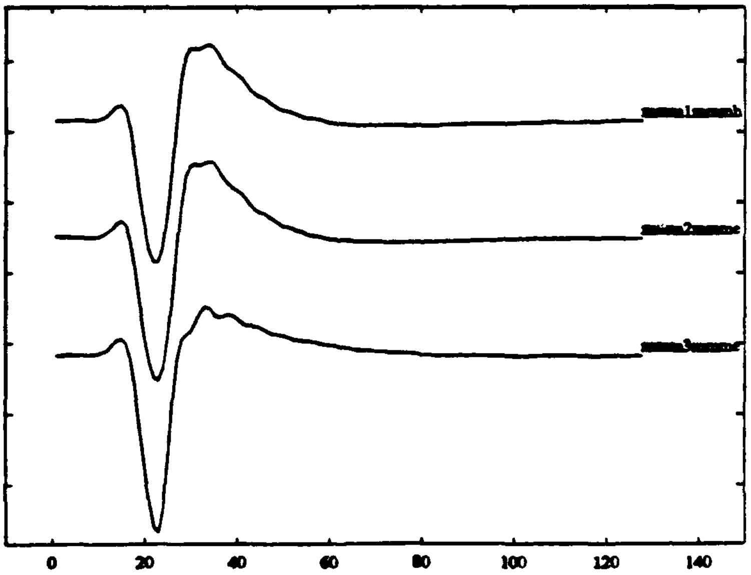

Figure 3.29: Group 4, (Signals 17-19).

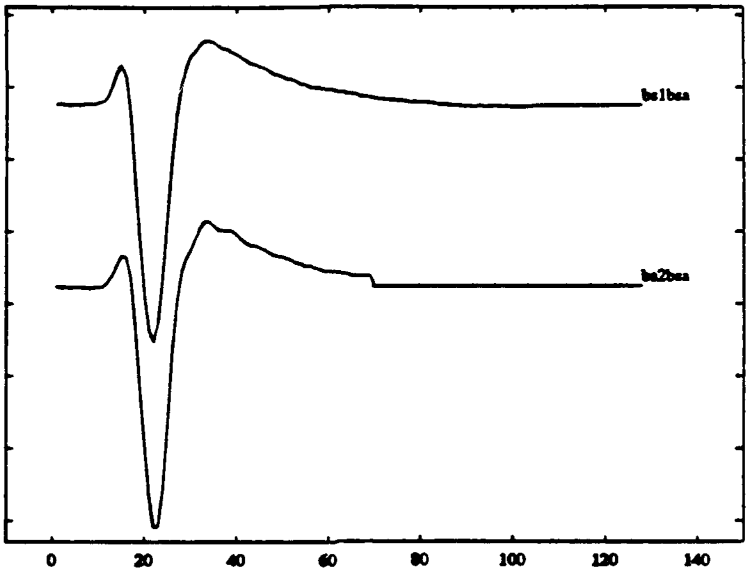

Figure 3.30: Group 5, (Signals 20-21).

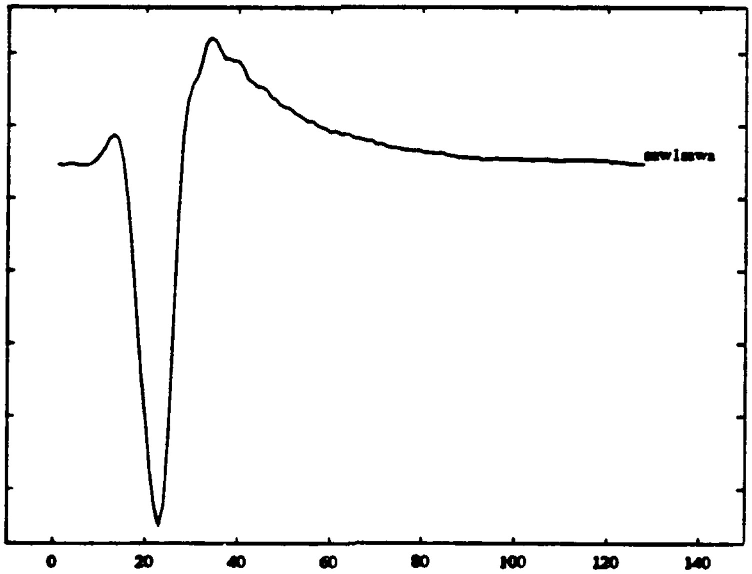

Figure 3.31: Group 6, (Signal 22).

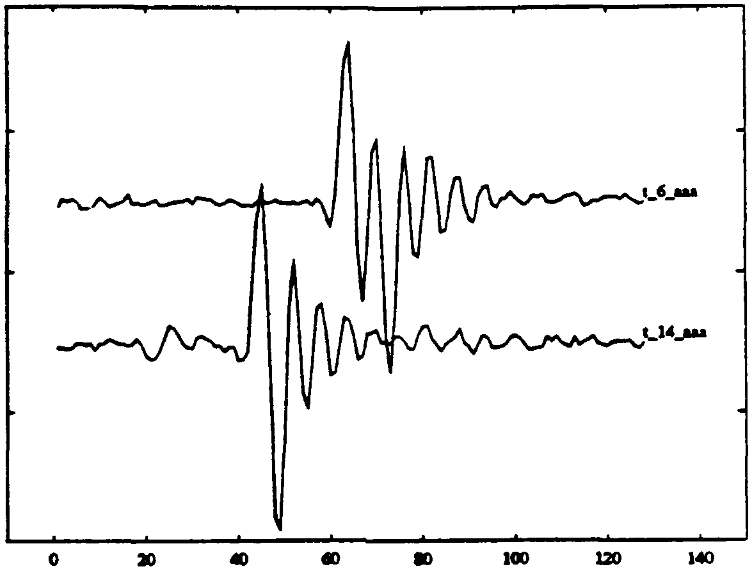

Figure 3.32: Group 7, (Signals 22-23).

3.2 Wavelet Basis Functions and Transform Topologies

This study considered rank 2 Mallat type wavelet transforms as well as rank 3 and 4 uniform type wavelet transforms. The signals considered in this project had relatively wide bandwidth which suggested an octave (Mallat) type transform. The Mallat type transformations proved to be superior early in the study and work on higher rank transforms was terminated after a few weeks.

The source signals were relatively smooth functions embedded in near Gaussian noise. We selected the Daubechies wavelets due to their optimal smoothness/vanishing moment characteristics. Wavelets from genus 2 through genus 10 were tested on the project data. The shorter systems (genera 2 and 3), seemed to be too susceptible to noise in the data. Genus 4 wavelets had good time resolution without sensitivity to noise. Genera 5 and greater tended to smear details due to the length of the filters.

Daubechies genus 4 wavelets were selected for this study. This wavelet system was used in a 6-level Mallat type transform topology.

3.3 Signal Processing Algorithms

This section presents a description of the prototype algorithms and the results of signal processing experiments conducted by Aware on the transient data to investigate the capabilities of wavelet based feature extraction methods. Of particular interest was the demonstration of the *feasibility* of performing feature extraction with a wavelet based approach and the *robustness* of the method with respect to natural signal variations and noise.

3.3.1 Detection Algorithm

All the transients in this study have strong cross-band time-frequency features that were exploited to detect the signals. In addition, most of the signals deviate from zero for a relatively long period of time. This characteristic was used early in the study to facilitate detection, but was abandoned later in the study when the full set of signals were considered. There were two signal detection algorithms used in the course of this study:

1. Zero-crossing method with local energy detector;

2. Cross-band wavelet method with peak-to-rms detector.

The detected transient was segmented into a 128 sample sub-record by including the ten samples prior to the leading edge of the detected transient and enough samples after that point to total 128 points.

3.3.2 Feature Extraction and Classification Algorithm

The feature extraction algorithm consisted of a continuous over-sampled wavelet transform based on the Daubechies length 8 (genus 4) wavelet. Six levels of transform were applied, resulting in an octave band analysis of the input data consisting of seven distinct bands (one low pass and six high pass octaves). This analysis bank is depicted in Figure 3.33, where, for the purpose of comparison two signals each from the first two groups were fed through the filtering structure.

Shift Invariance and oversampling

The goal of the classifier portion of the contract was to design an algorithm which was capable of distinguishing among the seven groups of example .ignals provided for the study. Examination of the signals and the results of the previous experiment suggested several important simplifying assumptions and discriminating features of this particular class of signals:

1. The transients as extracted by the detection algorithm are highly nonstationary.
2. Reliably grouping known classes requires an insensitivity to the time alignment of the transient within the selected time window.
3. Differences between classes tend to be small in terms of spectral or auto-correlative measures.

In order to both accomodate and exploit these characteristics the classification algorithm was designed to incorporate a distance measure which would be: shift invariant (and therefore unprejudiced to the inital alignment of the transients), finely grained in the time dimension (over-sampled), and based on coherent differences between instances of the transients.

Expanding on ideas originated by Mallat [4] on the use of octave band filtering in singularity classification, we chose a reduced representation of the input signal consisting of local extrema within each subband. Thus each subband output in the filter tree is replaced by an impulsive signal which is only nonzero at the locations where the subband had an extreme value and is equal to the original subband at those locations. In this representation it is easy to see the similarities between the reduced representation for each pair of signals and the differences between them. The initial representation for the signal is therefore a list of extrema for each subband along with the time location for each peak:

$$\begin{aligned}\text{Transient Feature Vector } =& \\ &\{F_i = (B(F)_i, X(F)_i, V(F)_i) \mid 1 \le i \le \text{number of peaks}\}\end{aligned}$$

where

$$\begin{aligned} B(F)_i &= \text{Band number for peak } \#i. \\ X(F)_i &= \text{Temporal location for peak } \#i. \\ V(F)_i &= \text{The value of peak } \#i. \end{aligned}$$

Figure 3.34 illustrates this reduced representation for a subset of example signals.

In order to further reduce to dimensionality of the signal representation, these peaks were pruned to only the six largest of the set. Thus the final representation of the signal prior to the computation of a pairwise distance function is a list of six peak values, their band numbers and their locations in time.

The first step in computing a pairwise distance function for these lists of peaks is to perform an alignment of the lists. By this we mean a revised list of peaks for each transient such that the number of peaks within each band is the same in both lists (each is both the same) and time ordering is preserved within each band. Such an alignment provides an initial indication of the similarity of two transients (whether the distibution of peaks among the bands are similar or not) and arranges the data conveniently for later analysis stages. Once the alignment is accomplished, peaks in one list are each naturally associated with the peak in the corresponding location in the other list.

The first method used to perform this alignment was based on excision. Peaks in each band were removed until the number of peaks within each band was the same for both lists. In each band, an interval of adjacent (sequential) peaks was chosen from the list with the greater number of peaks, and these were associated pairwise with the peaks in the list with fewer peaks. The remaining peaks in the list were removed from the calculation. The interval was chosen by minimizing the Euclidean distance between a peaks forming a given interval and the peaks in the other list. The temporal location of peaks was not used except to provide ordering of the peaks within a band. The distance assigned to the pair of lists is then the sum over the Euclidean distances in each band. Thus

$$d1(F_1, F_2) = (\sum_{(i,j) \in \text{matched peaks}} (V_{1,i} - V_{2,j})^2)^{1/2} \tag{3.1}$$

This scheme is in some sense a worst case distance function. It uses a minimal amount of information (a maximum of 6 values, the peak values themselves, are used to identify a signal), and provides for no penalty for the peaks which do not match in the chosen alignment; these were simply removed from the calulation.

The results of this algorithm in separating the various transients into their respective groups are shown in Figure 3.3.2. It is easily seen that even this simple approach succeeds in placing related transients at small distances from one another (within a group, all pairs should be enclosed by a single contour), and is also effective at separating the various groups from one another. However, the decision levels that arise from this algorithm are not very well separated, i.e., the distances between groups are of the same order of magnitude as those within groups.

The second method attempted to improve on this situation by incorporating all of the data from the pruned peak lists. This method uses the temporal location information to improve the match and computes a distance function which weights the contribution of a given peak to peak distance by the distance between the temporal locations of the peaks. For this purpose, the mean of the temporal data is removed, thereby making these distances into relative delays between the various peaks.

The other improvement over method 1 is to insert peaks in a list with fewer peaks in a given band, rather than removing them from the list with the greater number. This has the effect of adding a penalty to the distance function whenever a peak of significant magnitude in one list cannot be matched with a corresponding peak in the other list.

The alignment technique for this method is essentially the same as for the first method, except that peaks of zero magnitude are inserted when there is a mismatch, as opposed to the excision exercised by method 1. The distance function is then

$$d2(F_1, F_2) = (\sum_{(i,j) \in \text{matched peaks}} (|X2, i - X_{2,j}|(V_{1,i} - V_{2,j})^2)^{1/2} \qquad (3.2)$$

The results of the second algorithm in separating the various transients into their respective groups are shown in Figure 3.3.2. This approach is very successful in placing related transients at small distances from one another (within a group, all pairs should be enclosed by a single contour), and is also effective at separating the various groups from one another. The performance in this situation is *perfect* if one considers that signal 10 seems to be misgrouped.

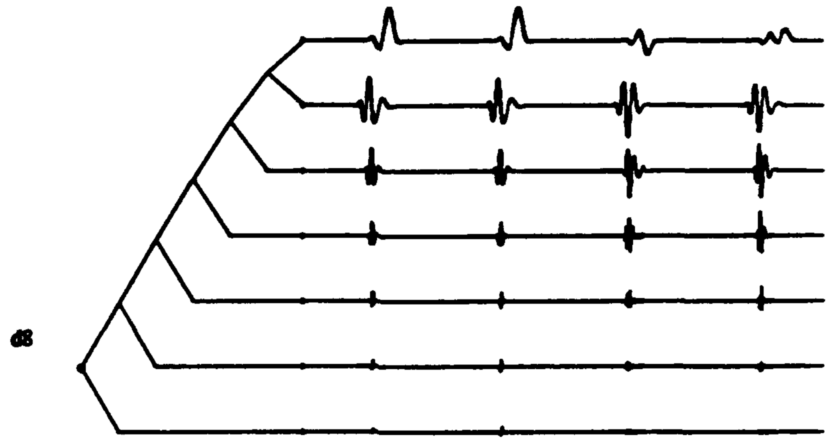

Figure 3.33: Over sampled wavelet transform of the first four example transients

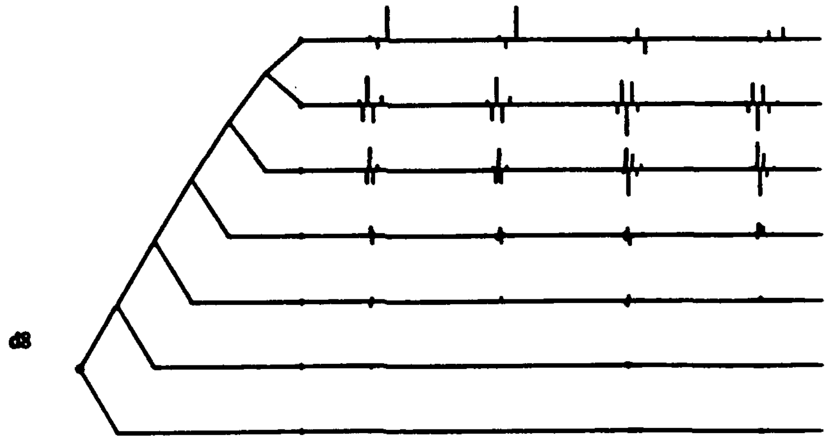

Figure 3.34: Local extrema for the first four example transients

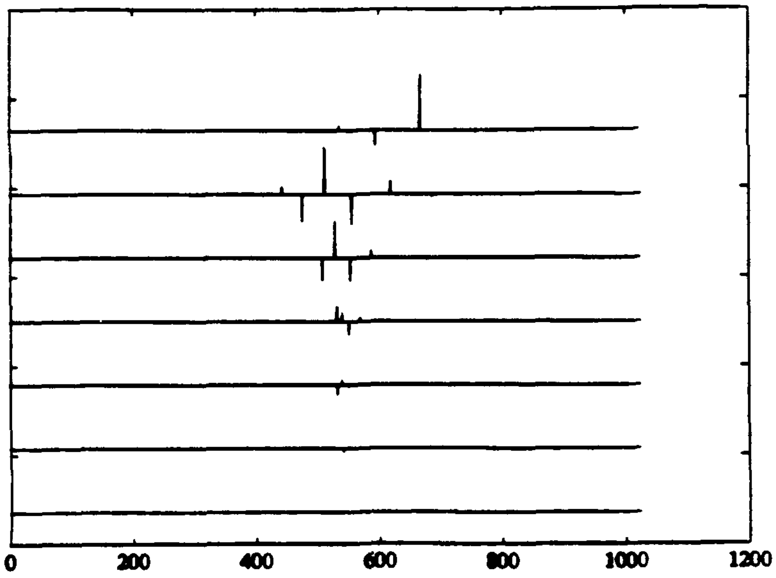

Figure 3.35: Local extrema for transient 1

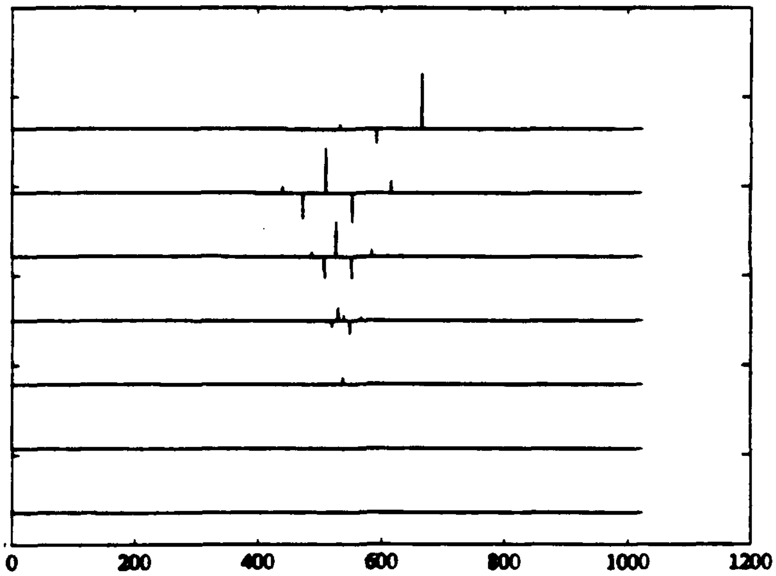

Figure 3.36: Local extrema for transient 2

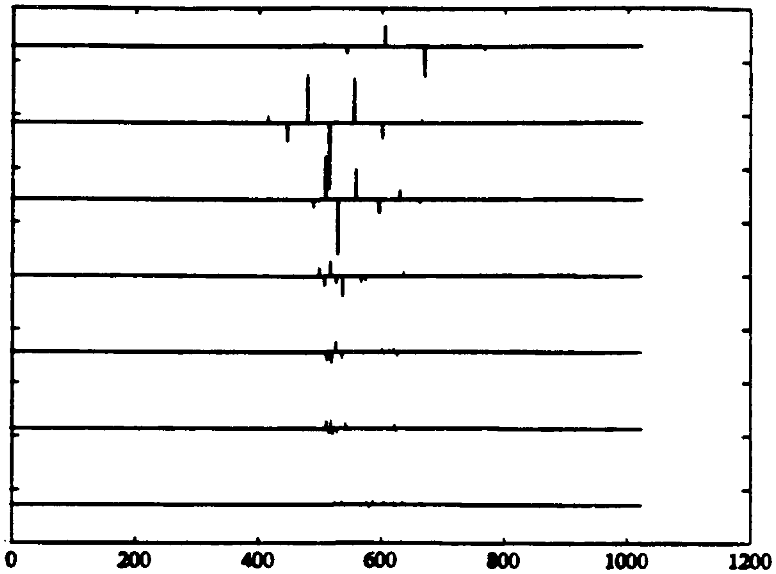

Figure 3.37: Local extrema for transient 3

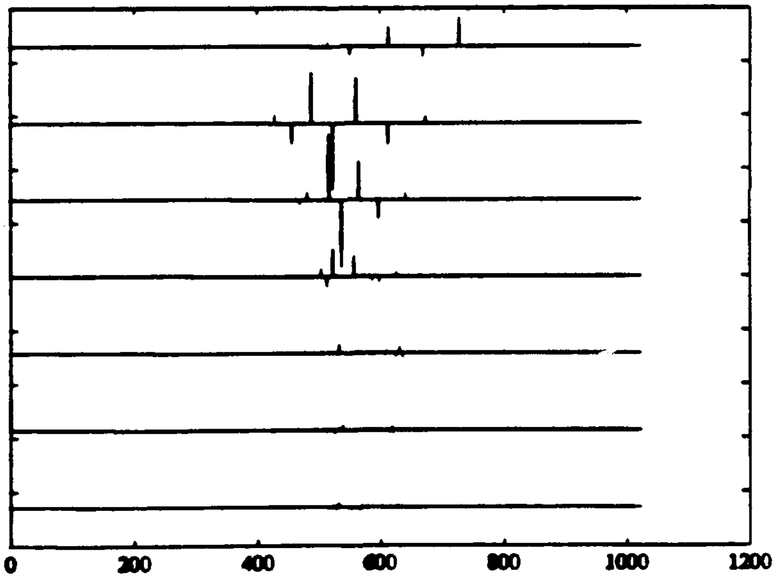

Figure 3.38: Local extrema for transient 4

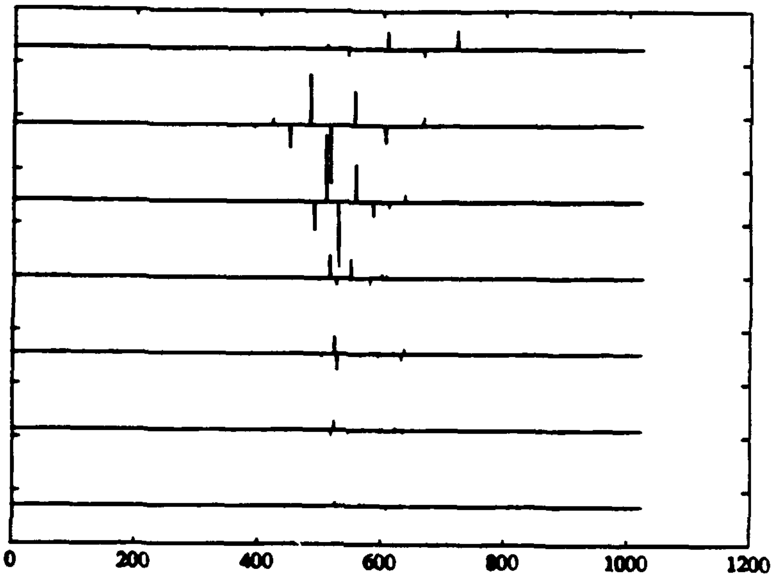

Figure 3.39: Local extrema for transient 5

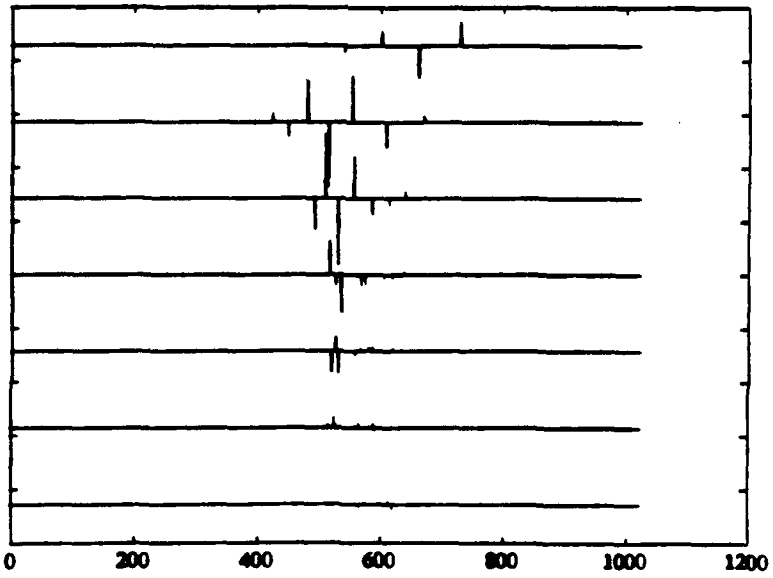

Figure 3.40: Local extrema for transient 6

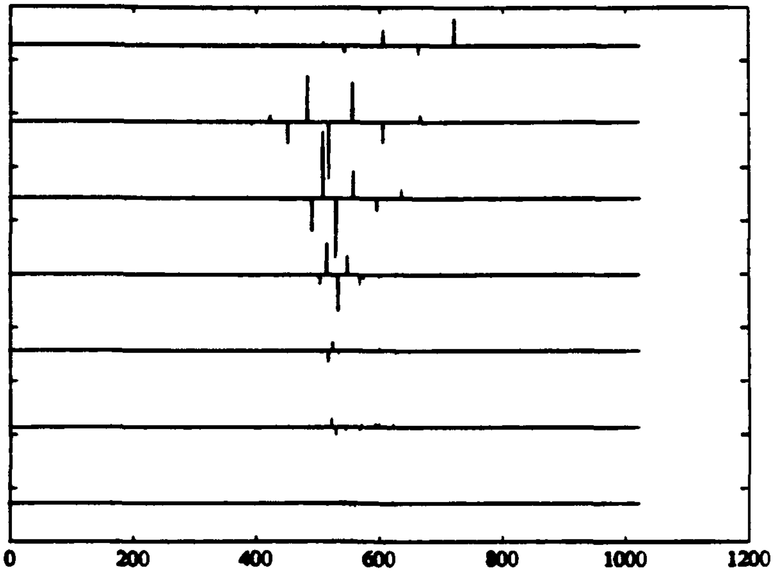

Figure 3.41: Local extrema for transient 7

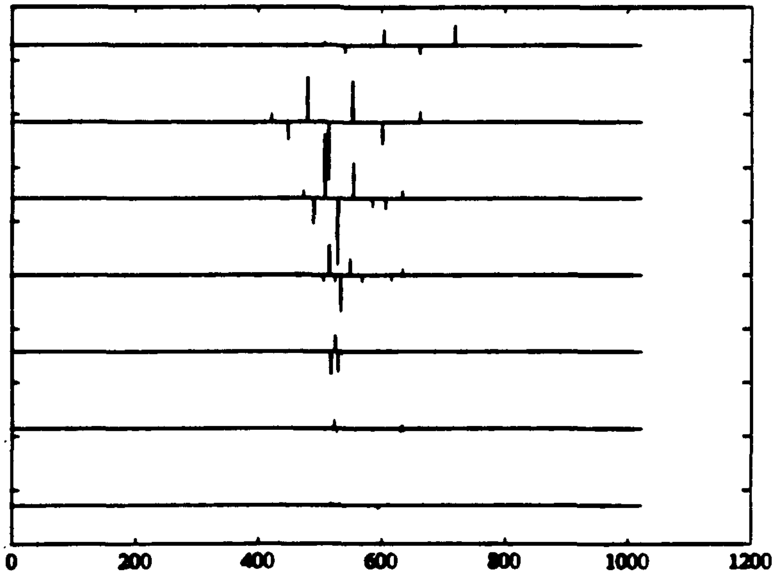

Figure 3.42: Local extrema for transient 8

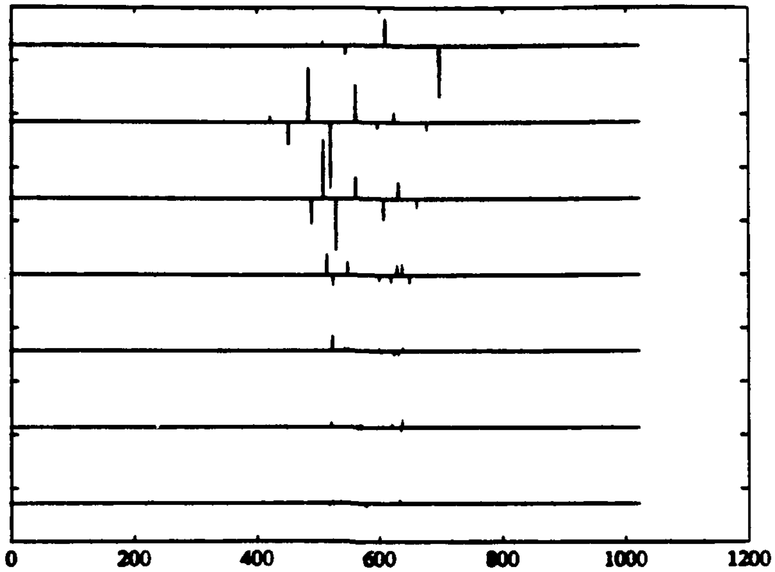

Figure 3.43: Local extrema for transient 9

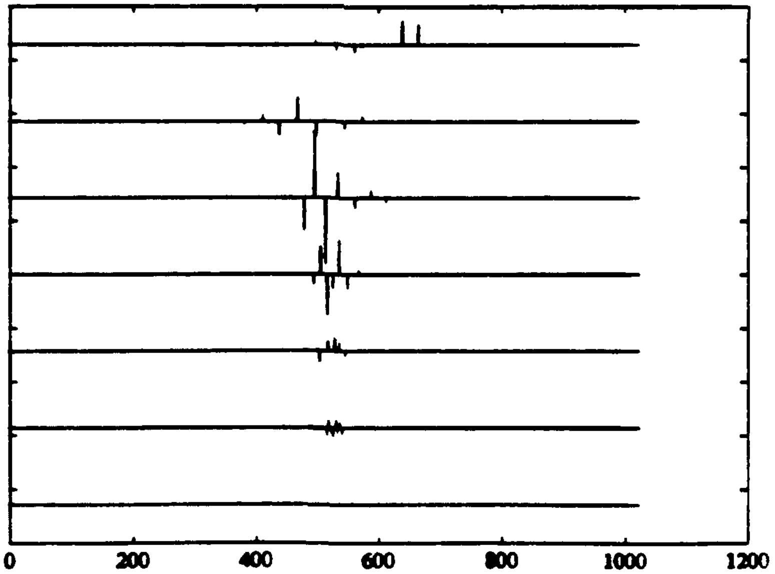

Figure 3.44: Local extrema for transient 10

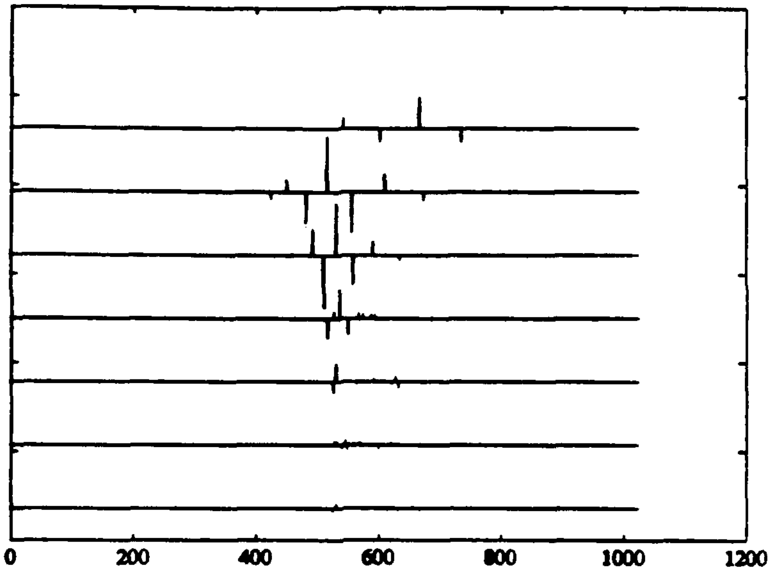

Figure 3.45: Local extrema for transient 11

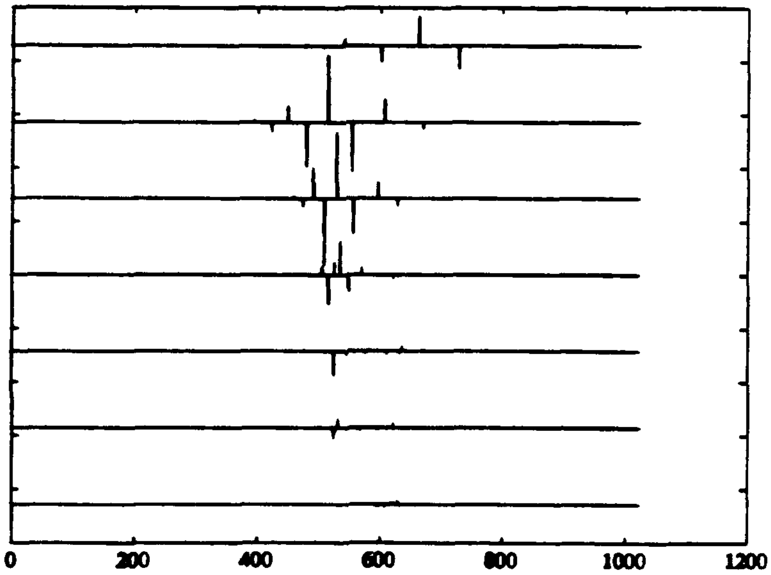

Figure 3.46: Local extrema for transient 12

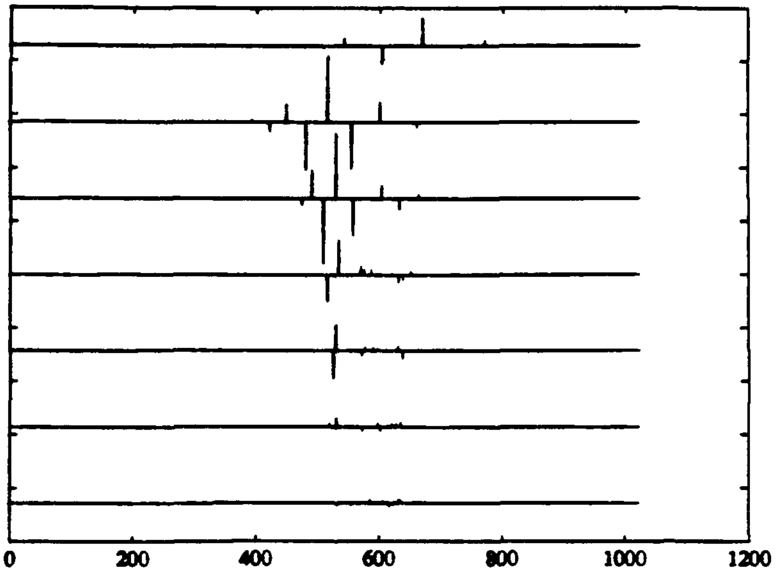

Figure 3.47: Local extrema for transient 13

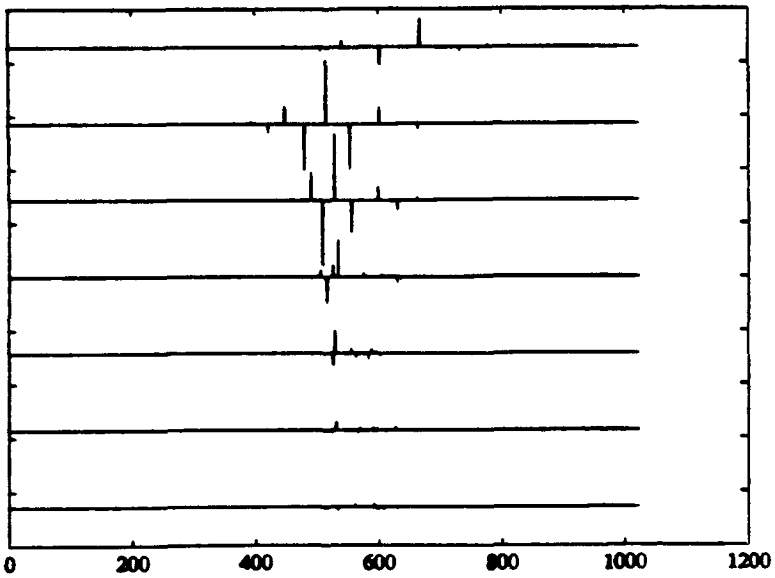

Figure 3.48: Local extrema for transient 14

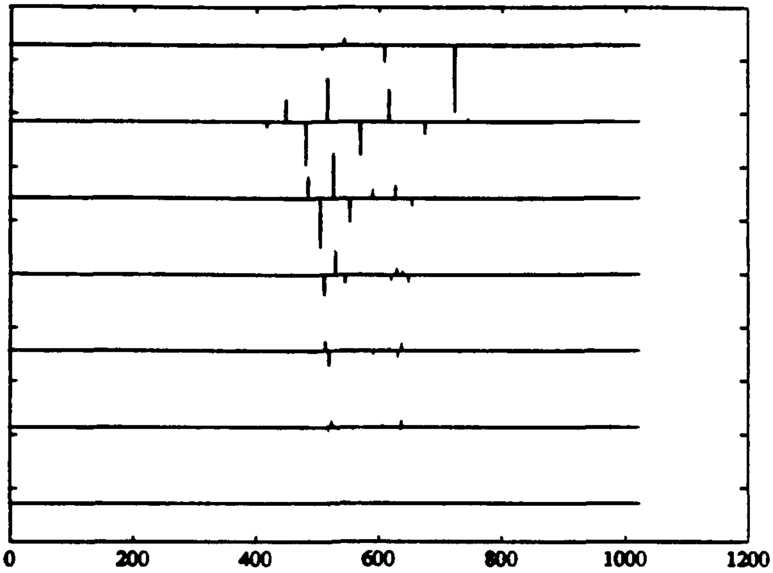

Figure 3.49: Local extrema for transient 15

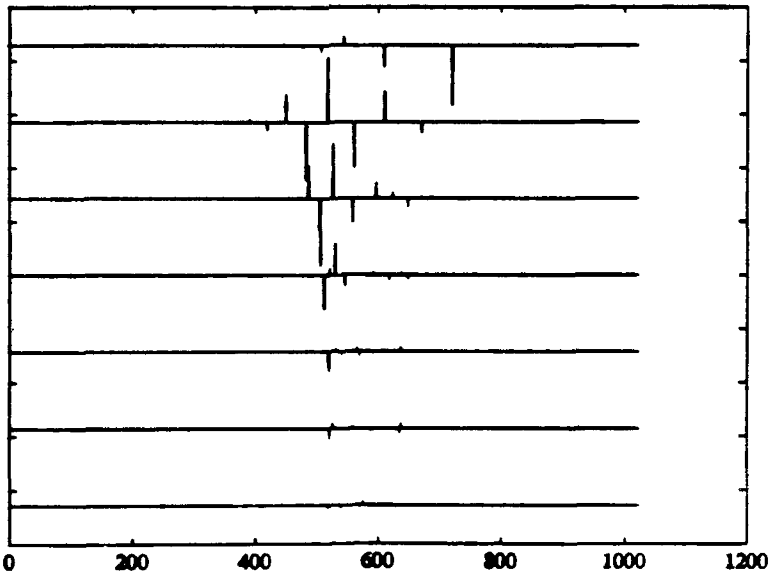

Figure 3.50: Local extrema for transient 16

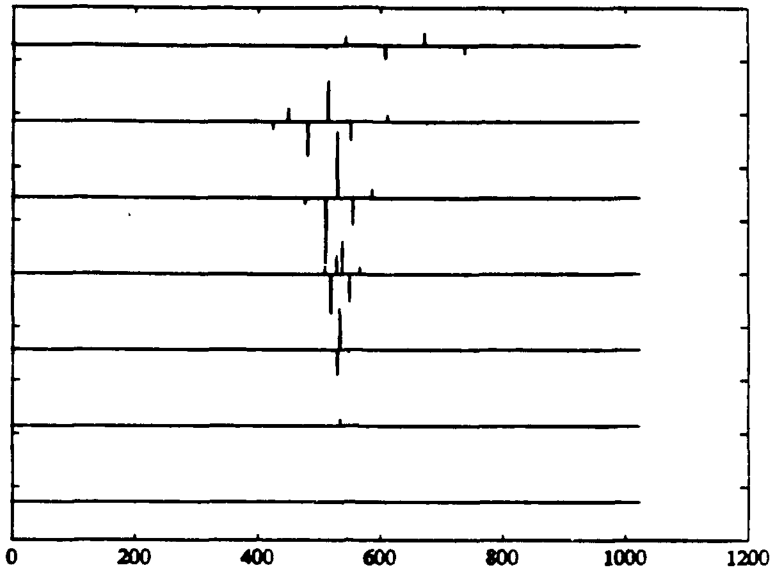

Figure 3.51: Local extrema for transient 17

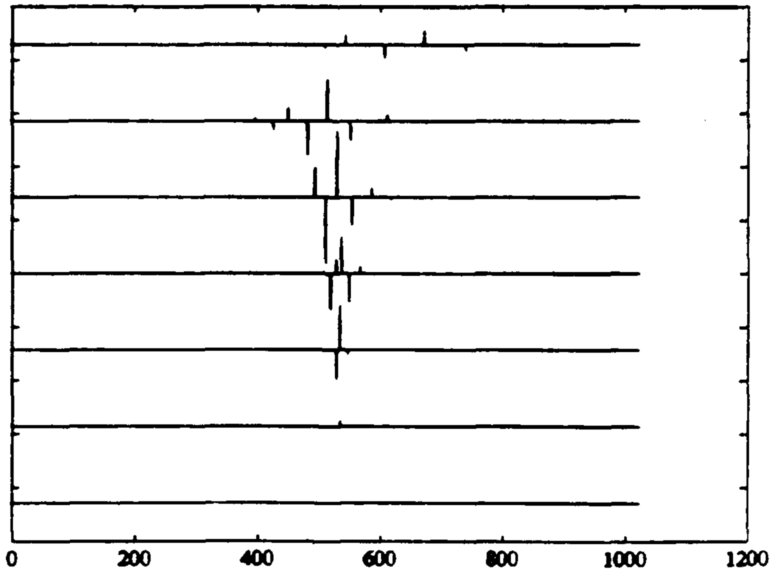

Figure 3.52: Local extrema for transient 18

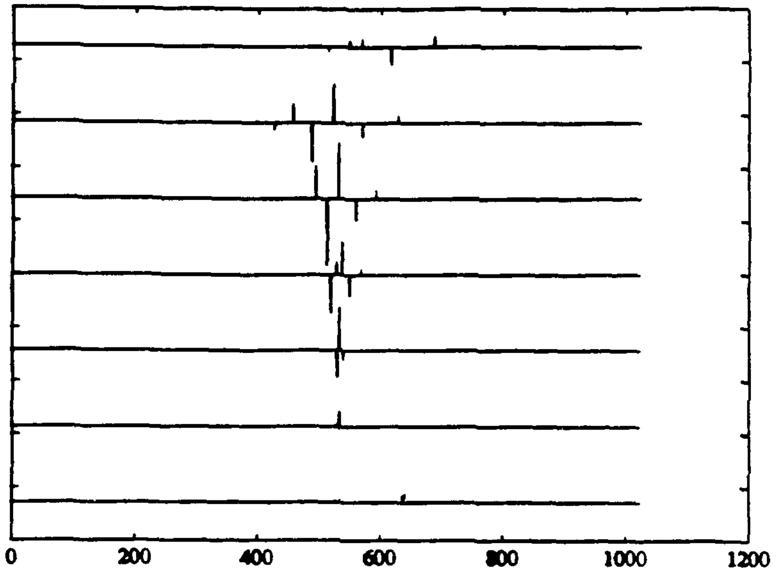

Figure 3.53: Local extrema for transient 19

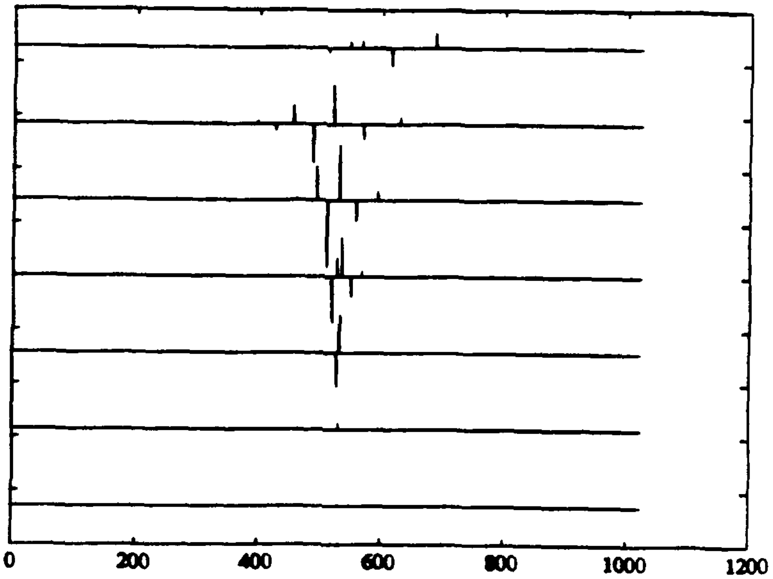

Figure 3.54: Local extrema for transient 20

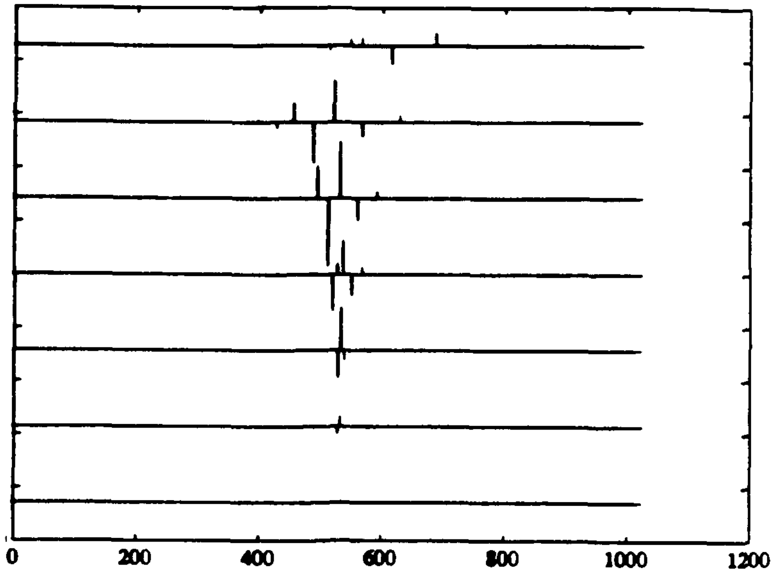

Figure 3.55: Local extrema for transient 21

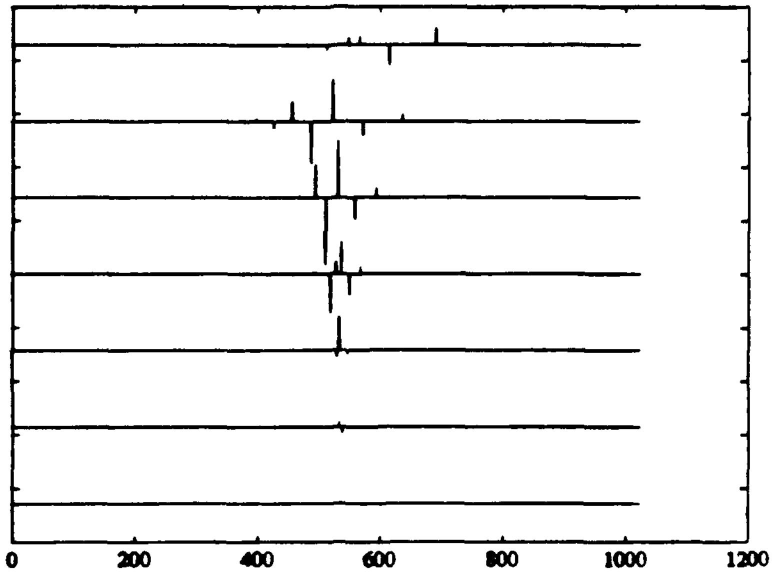

Figure 3.56: Local extrema for transient 22

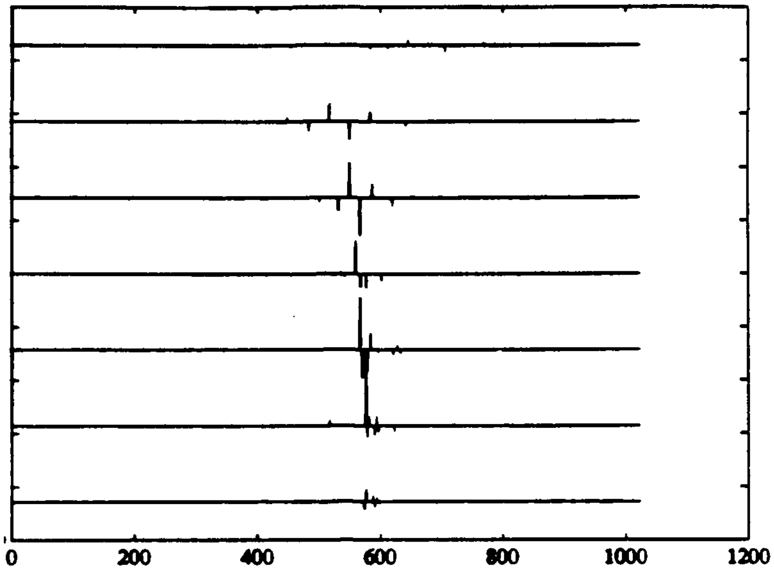

Figure 3.57: Local extrema for transient 23

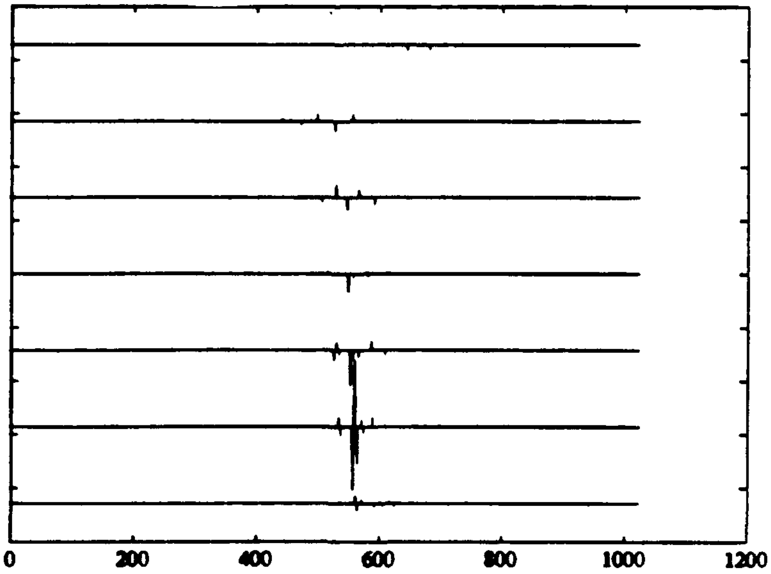

Figure 3.58: Local extrema for transient 24

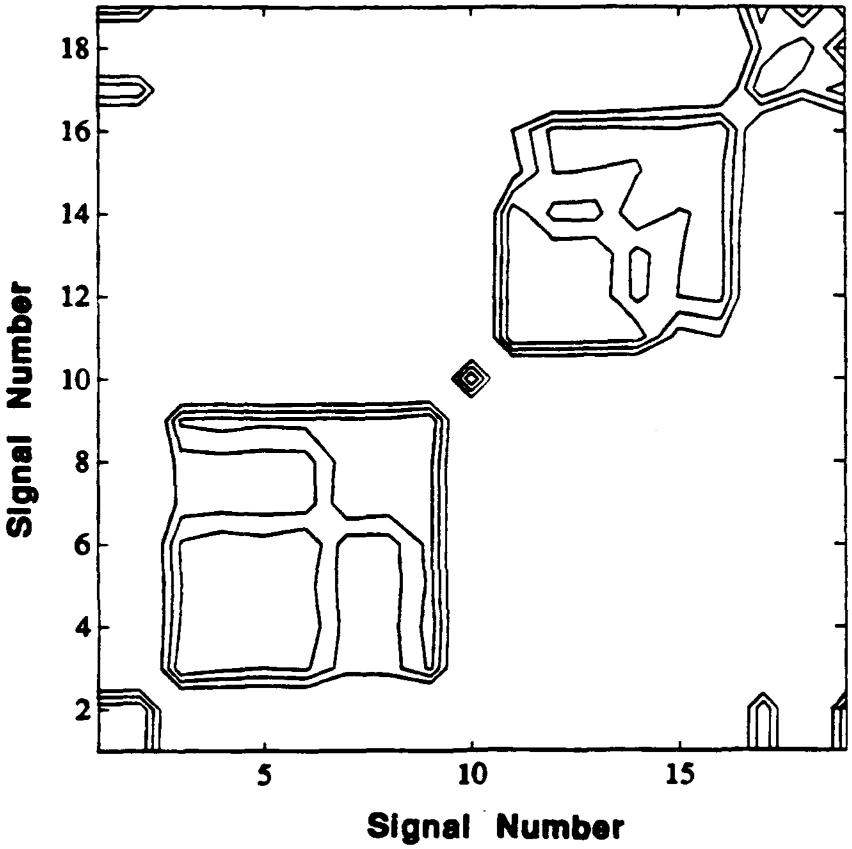

Figure 3.59: Correlation Surface for Distance Method 1

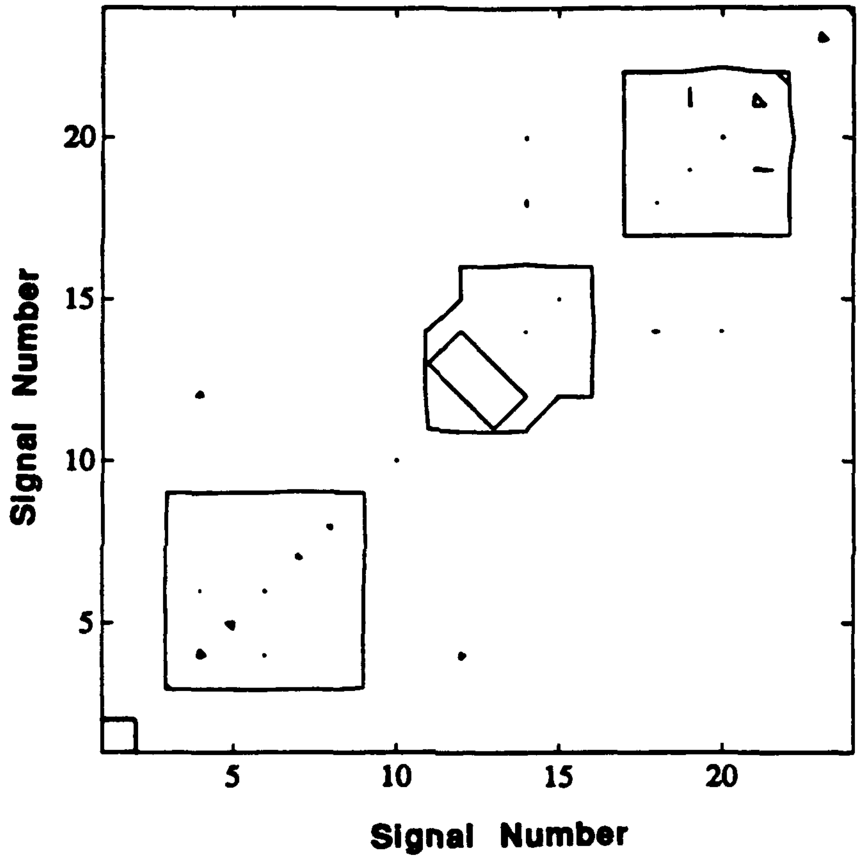

Figure 3.60: Correlation Surface for Distance Method 2

Chapter 4

Conclusion and Recommendations

This project demonstrated the feasibility and utility of employing wavelet transform based methods in the transient signal detection and feature extraction problem. Wavelet based transient signal analysis methods were shown to have the following desirable properties:

- Robust transient detection in the presence of strong sinusoidal noise components;
- Compact signal representation that allows a simple (low dimensional) classifier design to achieve near perfect signal separation; and
- Low computational complexity.

The logical 'next steps' required to develop this technology are:

1. Continue development on a larger class of transient signals. A question of particular importance is how the dimensionality of the classifier grows as a function of the size of the problem under consideration.
2. Development of *adaptive* versions of these signal processing algorithms would enable the fielding of signal analysis systems that could react to the observed environment.
3. Development of a high speed implementation of this technology would allow rapid progress in the research arena as well as enabling the technology to be demonstrated on a meaningful problem 'in the field.'

4. Development of hybrid classification systems that mate wavelet feature extraction with neural networks or other state-of-the-art classifier designs. There are many open questions regarding the requirements for the feature extraction and classifier portions of these high-performance hybrids.

Appendix A

List of Symbols

$\mathbf{D}$:= Ring of dyadic rational numbers
:= $\{x : x = m/2^n\ , \quad m, n \in \mathbf{Z}\ , n \geq 0\}$

$\mathbf{R}$:= Field of real numbers.

$\mathbf{Z}$:= Ring of rational integers.

μ := Rank for wavelet system. $\mu \in \mathbf{Z}$ and $\mu \geq 2$.

$[a]$:= Wavelet Coefficient Matrix ("WCM").
:= $\begin{pmatrix} a_0^0 & a_1^0 & \cdots & a_{N-1}^0 \\ a_0^{\mu-1} & a_1^0 & \ldots & a_{N-1}^{\mu-1} \end{pmatrix}$

where N is an integer multiple of μ ($N = g\mu$ where g is called the *genus*).

If $\mu = 2$ then we write:

$[a]$:= Wavelet Coefficient Matrix ("WCM").
:= $\begin{pmatrix} a_0 & a_1 & \cdots & a_{N-1} \\ b_0 & b_1 & \ldots & b_{N-1} \end{pmatrix}$

N := Number of columns in a WCM. N is an integer multiple of μ i.e. $g\mu$.
If $\mu = 2$ then N is eve

Appendix B

Wavelet Transform Theory

B.1 Wavelets and Wavelet Transforms

Most signals in science and engineering are modeled as mathematical functions for purposes of analysis. In order to separate or examine certain important features or characteristics of the signal, the function is often expanded in terms of basis functions that span the space or a subspace that the signals of interest reside in. The most common example of this is the Fourier transform where a signal that originates in the time domain is reformulated in the frequency domain by expanding the function in terms of trigonometric or complex exponential basis functions. This basis is most appropriate when the signals have periodic components or are produced by systems that are modeled by constant coefficient differential or difference equations.

Consider a periodic, possibly complex–valued, signal $g(t)$ that is square integrable over the range $\{0 \leq t \leq 1\}$ and with period one so that

$$g(t) = g(t+1). \tag{B.1}$$

This function can be expanded in a Fourier series of the form

$$g(t) = \sum_{n=-\infty}^{\infty} b_n \, e^{i2\pi nt} \tag{B.2}$$

with the coefficients given by

$$b_n = \int_0^1 g(t) \, e^{-i2\pi nt} \, dt \tag{B.3}$$

which is an inner product of $g(t)$ with the basis functions. Similarly, one defines the wavelet transform with respect to a basis of wavelet functions.

The wavelet basis is generated by a $\mu \times N$ matrix $[a]$, where μ and N are positive integers and N is a multiple of μ. The multiplier g is called the *genus* of the system and $N = g\mu$. The matrix

$$[a] := \begin{pmatrix} a_0^0, & \ldots, & a_{N-1}^0 \\ \vdots & a_j^i & \vdots \\ a_0^{\mu-1} & \ldots & a_{N-1}^{\mu-1} \end{pmatrix} \tag{B.4}$$

is a *wavelet coefficient matrix* ("WCM") if it satisfies the *scaling conditions*

$$\sum_{k=0}^{N-1} \overline{a}_k^i a_{k+\mu l}^j = \mu \delta_{i,j} \delta_{0,l} \tag{B.5}$$

$$\sum_k a_k^i = \mu \delta_{0,i} \tag{B.6}$$

where $\delta_{i,j}$ equals 1 if $i = j$ and 0 otherwise. The overbar denotes complex conjugation and l is an integer. The sums over k are finite sums since only finitely many of the numbers a_k^i are different from zero.

The positive integer μ is called the *multiplier* of the wavelet system and N is called its *length*;

This matrix of numbers provides coefficients for the vector of recursions

$$\varphi^i[a](x) = \sum_{k=0}^{N-1} a_k^i \varphi^0[a](2x - k) \tag{B.7}$$

which implicitly define the *wavelet scaling function* $\varphi^0[a]$ and explicitly define the *basic wavelet functions* $\varphi^i[a](x)$, $1 \leq i < \mu$. Observe that only $\varphi^0[a]$ appears on the right hand side. The functions $\varphi^i[a](x)$ are defined for all real numbers $x \in \mathbf{R}$.

The fundamental fact about systems of compactly supported wavelets is that the collection of functions

$$\mathbf{Basis}[a] := \Big\{ \mu^{j/2} \varphi^i[a](\mu^j x - k) :$$

$$0 \leq i < \mu, \quad j,k, \in \mathbf{Z}\}$$

form a basis for L^2 spaces, in particular for $L^2(\mathbf{R})$.

We now focus on the case of $\mu = 2$ and will use the following simplified notation for the scaling function $\varphi(t) \equiv \varphi^0[a](t)$ and the wavelet function $\psi(t) \equiv \varphi^1[a](t)$ where $t \in \mathbf{R}$. It can be shown that if the coefficients of this equation satisfy the wavelet conditions, stated above, the solution $\varphi(t)$ will be orthogonal to integer translates of itself and can be normalized such that

$$< \varphi(t), \varphi(t-k) > \triangleq \int \varphi(t)\,\varphi(t-k)\,dt = \delta_{0,k} \tag{B.8}$$

This means the set of basis functions

$$\varphi_l(t) = \varphi(t-l) \tag{B.9}$$

spans a subspace $\mathcal{V}_0$ in L^2 and the coefficients of an expansion within this subspace can be calculated as simple inner products. The feature of scaling functions that makes them attractive for signal processing is their ability to model signal properties that are related to the independent variable t. One can increase the size of the subspace spanned by the scaling functions by using $\varphi_{j,k}(t) := 2^{j/2}\varphi(2^j t - k)$ which spans a subspace $\mathcal{V}_j$. One can show that $\mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \cdots$.

The features of a signal can often be better described by defining a slightly different set of orthogonal basis functions that span the *differences* between the spaces spanned by the various scales of the scaling function. These new functions are the wavelets. The ***basic wavelet*** is defined in terms of the scaling function by

$$\psi(t) = \sum_k (-1)^k a_{N-k-1}\,\varphi(2t-k)\,. \tag{B.10}$$

It is the prototype of a class of orthonormal basis functions of the form

$$\psi_{j,k}(t) = 2^{j/2}\,\psi(2^j t - k) \tag{B.11}$$

where 2^j is the scaling of t. $2^{-j}k$ is the translation in t, and $2^{j/2}$ maintains the unity norm of the wavelet. We shall say that j is the base-2 logarithm of the scale. If $\mathcal{W}_j$ is the subspace of $L^2 = L^2(\mathbf{R})$ spanned by the integer translates of the wavelet $2^{j/2}\psi(2^j t)$, additional disjoint subspaces are spanned by integer translates of the wavelets for each different scale index j, that is, by the functions $\psi_{j,k}(t)$. The relationship of the various subspaces can be seen from the following expressions:

$$\mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \cdots \subset L^2 = L^2(\mathbf{R}) \,, \tag{B.12}$$

$$\mathcal{V}_0 \oplus \mathcal{W}_0 = \mathcal{V}_1 \,, \tag{B.13}$$

$$\mathcal{V}_{j-1} \oplus \mathcal{W}_{j-1} = \mathcal{V}_j \,, \tag{B.14}$$

$$L^2 = \mathcal{V}_0 \oplus \mathcal{W}_0 \oplus \mathcal{W}_1 \oplus \cdots \tag{B.15}$$

and indeed, if we allow j to run over *all* integers, then

$$L^2 = \ldots \mathcal{W}_{-2} \oplus \mathcal{W}_{-1} \oplus \mathcal{W}_0 \oplus \mathcal{W}_1 \ldots \mathcal{W}_j \ldots \,. \tag{B.16}$$

This states that the set of basis functions formed from $\varphi_l(t)$ and $\psi_{j,k}(t)$ span all of L^2 and, therefore, any function in L^2 can be written

$$g(t) = \sum_{l=-\infty}^{\infty} c_l \, \varphi_l(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k} \, \psi_{j,k}(t) \tag{B.17}$$

with the coefficients expressed by

$$c_l = \int g(t) \, \varphi_l(t) \, dt \tag{B.18}$$

and

$$d_{j,k} = \int g(t) \, \psi_{j,k}(t) \, dt. \tag{B.19}$$

The basis functions $\varphi_l(t)$ and $\psi_{jk}(t)$ are numerical valued functions of numerical variables; they have no physical dimension. In the expansion formula (B.17) the argument $2^j t - k$ of ψ is a pure number so, writing

$$2^j t - k = 2^j \left(t - \frac{k}{2^j} \right)$$

the quantity 2^j has the dimension t^{-1}, i.e. frequency, and $k/2^j$ has the dimension time.

These wavelet coefficients completely and uniquely describe the original signal and can be used to represent it in a way similar to Fourier coefficients. Because of the orthonormality of the basis functions, there is a version of Parseval's theorem that relates the energy of the signal $g(t)$ to the energy in each of the wavelet expansion components and their wavelet coefficients by the formula

$$||g||^2 = \sum_l |c_l|^2 + \sum_{j\geq 0} \sum_{k=-\infty}^{\infty} |d_{j,k}|^2 \,. \tag{B.20}$$

This is one reason why the orthonormality is so important. Daubechies [2] showed that the translates of the scaling function and the translated dilations of the wavelets are orthonormal, and all of the these functions have compact support (i.e. are non-zero only over a finite region) if there are only a finite number of non-zero coefficients a_k in the recursive scaling equation (B.7). This provides the time localization that is particularly desirable for analyzing both the time and the frequency behavior of transient signals.

Note that there is an infinite set of scaling functions and wavelets that can be obtained by choosing different coefficients a_k in (B.7).

B.2 Energy and Parseval's Theorem

Orthonormal basis systems allow direct calculation and interpretation of the energy in a signal partitioned in both the time and the expansion domains. Parseval's theorem for the Fourier series (B.2) states

$$\int_0^1 |g(t)|^2 \, dt = \sum_{n=-\infty}^{\infty} |b_n|^2 . \tag{B.21}$$

The "power" in a signal is proportional to the square of the signal (e.g. voltage, current, force, or velocity) and, therefore, the energy is given by the integral of the square of the signal magnitude. Parseval's theorem states how the total energy is partitioned in the frequency domain in terms of the partition provided by the the orthonormal basis functions. For the general wavelet expansion of (B.17), Parseval's theorem is

$$\int_0^1 |g(t)|^2 \, dt = \sum_{l=-\infty}^{\infty} |c_l|^2 + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} |d_{j,k})|^2 \tag{B.22}$$

with the energy in the expansion domain partitioned in time by l and k and in scale by j. For the case of periodic functions, the relationship reduces to

$$\int_0^1 |g(t)|^2 \, dt = |c_l|^2 + \sum_{j=0}^{\infty} \sum_{k=1}^{2^j} |d_{j,k})|^2 \,. \tag{B.23}$$

One can show that $|\hat{\psi}(f)| \rightarrow 0$ as $f \rightarrow 0$ and also as $f \rightarrow \infty$. Therefore, there will be a band of frequencies where most of the energy in $\hat{\psi}(f)$ is concentrated. Likewise, for many signals, the energy will be concentrated in a region of the (j,k) plane. Because of this concentration, the energy of the signal $g(t)$ at frequency n and at scale j and time k is approximately measured by

$$2^{-j}\left|\hat{\psi}\left(\frac{n}{2^j}\right)\right|^2 \; |d_{j,k}|^2 \tag{B.24}$$

If most of the energy in $\hat{\psi}(f)$ occurs around frequency f_0, then $f_0 = n/2^j$ relates the dominant Fourier frequency f_0 to the dominant wavelet scale j. Scale and frequency are independent primitive concepts, but the selection of a wavelet basis establishes a connection between them, and the results of this chapter allow one to move between the two descriptions, using the one most appropriate for a particular problem. The simple partitioning of the energy content of a signal among frequencies has been generalized to include the parameters of time and scale. In addition, we have at our disposal the choice of wavelet systems, which is controlled by the choice of a_k in (B.7), and determines the detailed nature of the relationship between frequency and scale.

Bibliography

[1] C. S. Burrus and T. W. Parks, *(DFT/FFT) and Convolution Algorithms*, John Wiley and Sons, New York, 1985.

[2] I. Daubechies, "Orthonormal bases of compactly supported wavelets", *Communications on Pure and Applied Mathematics* 1988, Vol. 41, 909-996.

[3] S. G. Mallat, "A Theory for multiresolution signal decomposition: the wavelet representation", *(IEEE) Transactions on Pattern Recognition and Machine Intelligence*, July 1989, Vol. 11, no. 7, 674-693.

[4] S. G. Mallat, W. L. Hwang, "Singularity Dectection and Processing with Wavelets", *(IEEE) Transactions on Information Theory*, March 1992, Vol. 38, no. 2, 617-643.